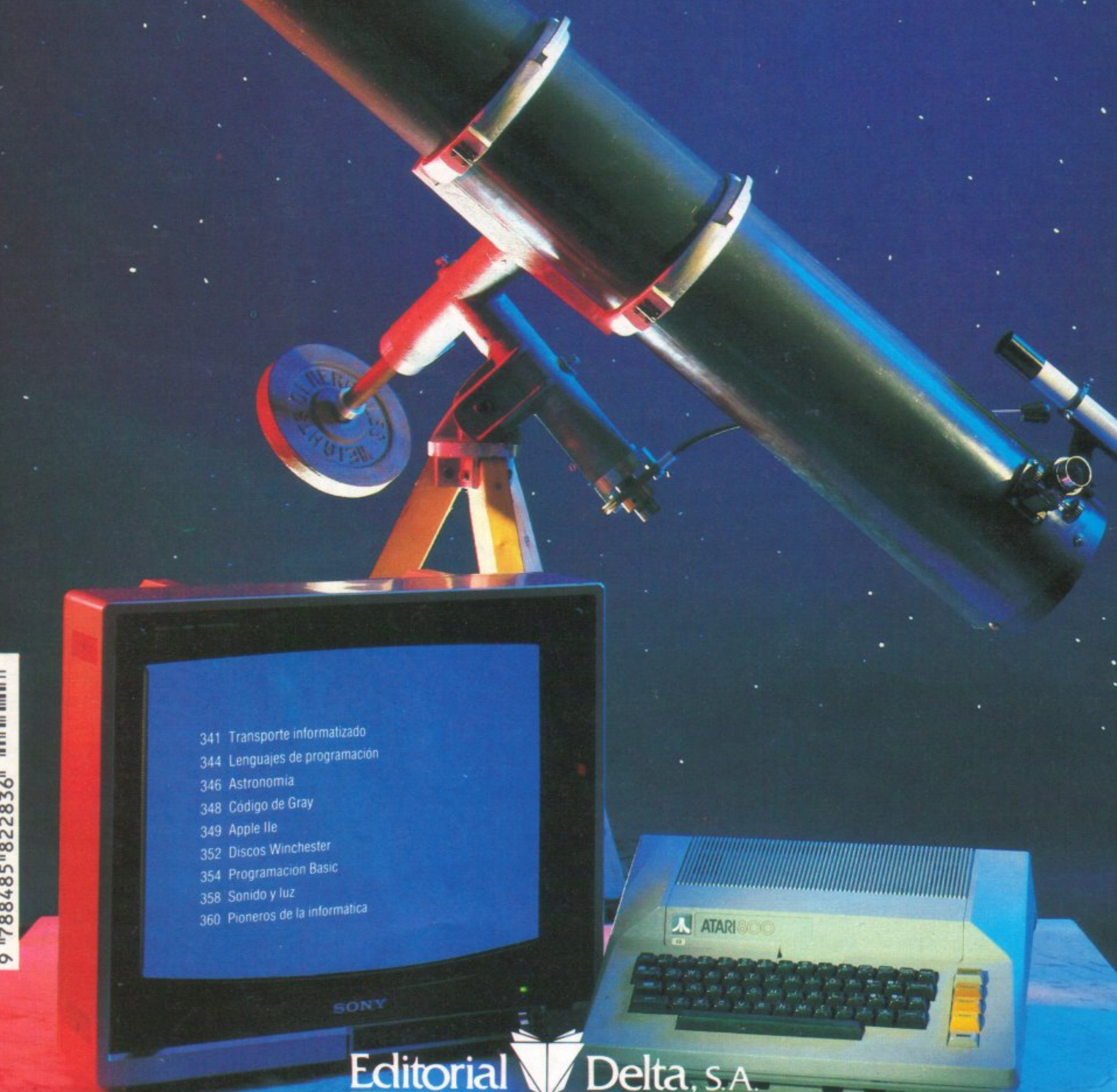
mi COMPUTER

18

CURSO PRACTICO DEL ORDENADOR PERSONAL, EL MICRO Y EL MINIORDENADOR



9 788485 822836 10018

Editorial Delta, S.A.

150 ptas.

MI COMPUTER CURSO PRACTICO
DEL ORDENADOR PERSONAL, EL MICRO Y EL MINIORDENADOR

Publicado por Editorial Delta, S.A., Barcelona, y comercializado en exclusiva por Distribuidora Olimpia, S.A., Barcelona

Volumen II - Fascículo 18

Director: José Mas Godayol
Director editorial: Gerardo Romero
Jefe de redacción: Pablo Parra
Coordinación editorial: Jaime Mardones
Asesor técnico: Jesús Nebra

Redactores y colaboradores: G. Jefferson, R. Ford, S. Tarditti, A. Cuevas
Para la edición inglesa: R. Pawson (editor), D. Tebbutt (consultant editor), C. Cooper (executive editor), D. Whelan (art editor), Bunch Partworks Ltd. (proyecto y realización)

Realización gráfica: Luis F. Balaguer

Redacción y administración:
Paseo de Gracia, 88, 5.°, Barcelona-8
Tels. (93) 215 10 32 / (93) 215 10 50 - Télex 97848 EDLTE

MI COMPUTER, *Curso práctico del ordenador personal, el micro y el miniordenador,* se publica en forma de 96 fascículos de aparición semanal, encuadernables en ocho volúmenes. Cada fascículo consta de 20 páginas interiores y sus correspondientes cubiertas. Con el fascículo que completa cada uno de los volúmenes, se ponen a la venta las tapas para su encuadernación.
El editor se reserva el derecho de modificar el precio de venta del fascículo en el transcurso de la obra, si las circunstancias del mercado así lo exigieran.

ISBN: 84-85822-83-8 (fascículo) 84-85822-90-0 (tomo 2)
84-85822-82-X (obra completa)
Depósito Legal: B. 52/1984

Fotocomposición: Tecfa, S.A., Pedro IV, 160, Barcelona-5
Impresión: Cayfosa, Santa Perpètua de Mogoda (Barcelona) 168405
Impreso en España - Printed in Spain - Abril 1984

Editorial Delta, S.A., garantiza la publicación de todos los fascículos que componen esta obra.

Distribuye para España: Marco Ibérica, Distribución de Ediciones, S.A., Carretera de Irún, km 13,350. Variante de Fuencarral, Madrid-34.
Distribuye para Argentina: Viscontea Distribuidora, S.C.A., La Rioja 1134/56, Buenos Aires.
Distribuye para Colombia: Distribuidoras Unidas, Ltda., Transversal 93, n.° 52-03, Bogotá D.E.
Distribuye para México: Distribuidora Intermex, S.A., Lucio Blanco, n.° 435, Col. San Juan Tlihuaca, Azcapotzalco, 02400, México D.F.
Distribuye para Venezuela: Distribuidora Continental, S.A., Ferrenquín a Cruz de Candelaria, 178, Caracas, y todas sus sucursales en el interior del país.

Pida a su proveedor habitual que le reserve un ejemplar de MI COMPUTER. Comprando su fascículo todas las semanas y en el mismo quiosco o librería, Vd. conseguirá un servicio más rápido, pues nos permite realizar la distribución a los puntos de venta con la mayor precisión.

Servicio de suscripciones y atrasados (sólo para España)

Las condiciones de suscripción a la obra completa (96 fascículos más las tapas, guardas y transferibles para la confección de los 8 volúmenes) son las siguientes:

a) Un pago único anticipado de 16 690 ptas. o bien 8 pagos trimestrales anticipados y consecutivos de 2 087 ptas. (sin gastos de envío).

b) Los pagos pueden hacerse efectivos mediante ingreso en la cuenta 3371872 de la Caja Postal de Ahorros y remitiendo a continuación el resguardo o su fotocopia a Distribuidora Olimpia (Paseo de Gracia, 88, 5.°, Barcelona-8), o también con talón bancario remitido a la misma dirección.

c) Se realizará un envío cada 12 semanas, compuesto de 12 fascículos y las tapas para encuadernarlos.

Los fascículos atrasados pueden adquirirse en el quiosco o librería habitual. También pueden recibirse por correo, con incremento del coste de envío, haciendo llegar su importe a Distribuidora Olimpia, en la forma establecida en el apartado b).

Para cualquier aclaración, telefonear al (93) 215 75 21.

No se efectúan envíos contra reembolso.



Transporte informatizado



Cortesía de YRM Arquitectos y Urbanistas

El "People Mover"
Siguiendo el ejemplo de muchos aeropuertos de Estados Unidos, se ha instalado en el aeropuerto de Gatwick, al sur de Londres, un revolucionario sistema de transporte interterminal; combina la estabilidad direccional y la capacidad de funcionar automáticamente, sin conductor, de los ferrocarriles ligeros, con la comodidad del autobús y el coche. Diseñado y construido por la Westinghouse Corporation, empresa famosa por sus sistemas de señalización de vías, el *People Mover* ("transportador de personas") puede trasladar hasta 100 pasajeros en cada viaje

En un mundo cada vez más poblado, transportar personas y bienes es una labor compleja. La utilización de ordenadores contribuye a que todo marche mejor

Hace 150 años la travesía desde Europa a Australia duraba tres meses. En la década de los ochenta, ese mismo viaje se podría efectuar en menos de doce horas. Sin embargo, este milagro tecnológico no habría sido posible de no existir métodos de control computerizado.

El viaje aéreo es la forma de transporte que plantea los problemas más serios. En el aeropuerto de Heathrow, en Londres, por ejemplo, se contabilizan más de un millar de vuelos diarios, con 120 desplazamientos de aviones por hora durante los períodos punta. Sin el auxilio de los ordenadores para controlar esta actividad, el sistema sencillamente no podría funcionar.

Consideremos el caso de un viajero que desee trasladarse de Londres a Nueva York. Desde la agencia de viajes, donde compraría el billete y se le reservaría una plaza (utilizando el Prestel como puerta para el ordenador de reservas de la propia compañía aérea), hasta que tocara tierra en el aeropuerto Kennedy, es posible que participen directamente en el viaje hasta 15 ordenadores diferentes. Analicemos con mayor detalle el papel del ordenador en un viaje aéreo.

Veamos primero el aparato. Los aviones de pasajeros modernos valen una enorme cantidad de dinero. Con el fin de garantizar al máximo su inversión, los operadores deben mantener al avión "como nuevo" en la mayor medida posible, y la clave para ello es el "mantenimiento planificado". Después de un número preestablecido de horas de vuelo, el aparato regresará a su base de ingeniería, donde el equipo de personal tendrá acceso a registros computerizados del historial completo del avión desde el día en que fue construido, pasando por el número de serie de cada pieza, tanto de las que están actualmente en uso como de las reemplazadas. Los registros detallarán todas las operaciones de ingeniería a las que haya sido sometido el avión; informes de los mecánicos de a bordo y de otros miembros de la tripulación acerca de su rendimiento; cifras del consumo de combustible, y cualquier otro tipo de dato que pueda considerarse de interés.

El avión sólo volverá a entrar en servicio cuando el plan de mantenimiento esté completo y actualizado. Y aun entonces se integrará a otro sistema informático: el sistema de control operativo de las líneas aéreas. Éste le señala al avión las rutas, distribuye pedidos de combustible en diversos puntos a través de esas rutas y se ocupa de la tripulación, las comidas, la atención y distracción de los pasajeros durante el vuelo y del gran número de medidas necesarias para transportar a trescientas o cuatrocientas personas a través de medio mundo.

Otro sistema de control computerizado opera dentro del propio aeropuerto, donde los funcionarios han de hacer frente a la inmensa demanda de sus recursos, a veces limitados, por parte de las compañías aéreas, para las que unos pocos minutos de retraso pueden representar una considerable pérdida de dinero. Esta operación sólo se puede

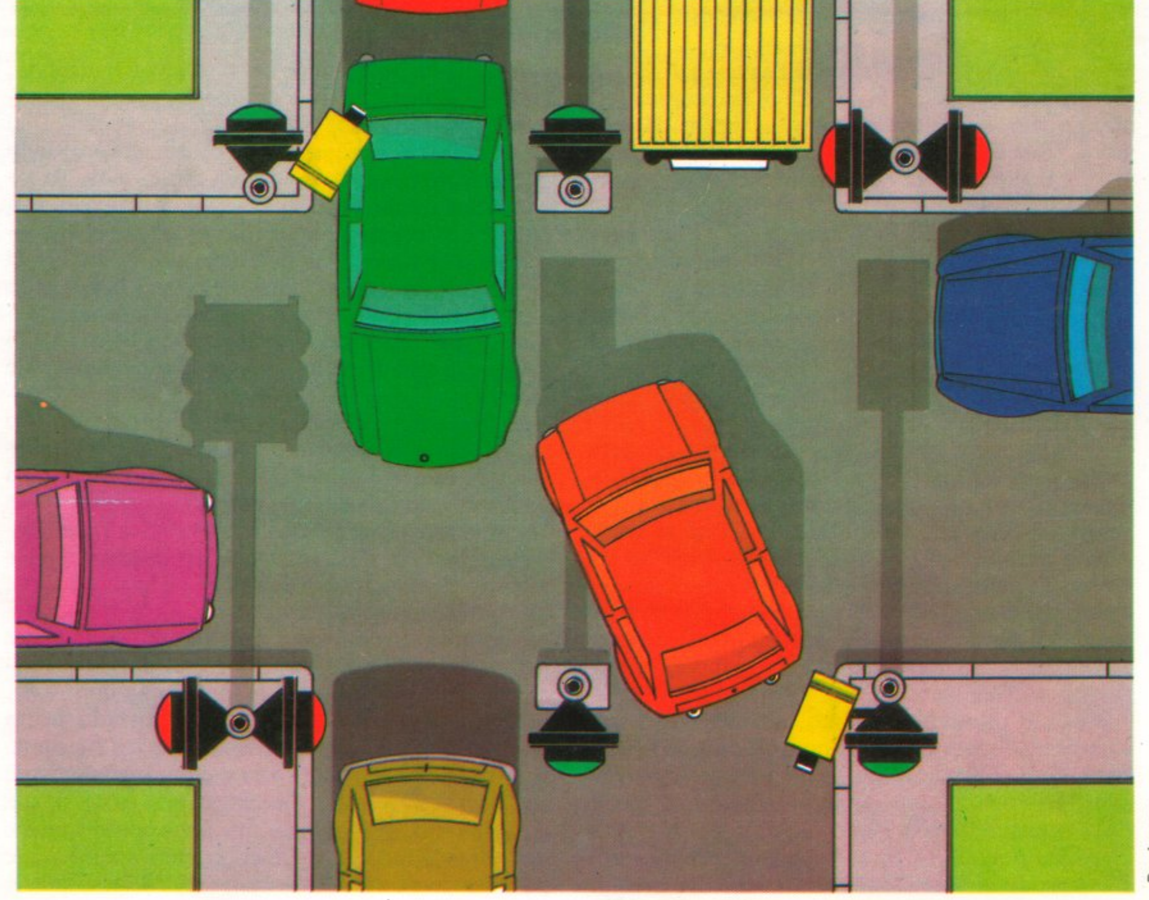
llevar a cabo con total éxito gracias a la planificación computerizada. El sistema informático del aeropuerto también se ocupará de llamar a los pasajeros para verificar las listas de vuelo y de visualizar la información relativa a salidas y llegadas.

Antes de que los pasajeros hayan siquiera llegado al aeropuerto, el piloto habrá archivado un detallado plan de vuelo con el Control de Tráfico Aéreo. De modo sorprendente, la tarea del CTA se efectúa sólo parcialmente asistida por ordenador. Gracias a los sistemas de radiofaro de respuesta por radar, los controladores ya no tienen que depender de que el piloto comunique su posición verbalmente. Cada blip que ellos ven en sus pantallas de radar se identifica mediante un número de vuelo, junto con la lectura de altitud interpretada por ordenador y el código de destino que se transmiten automáticamente desde el avión. El controlador dispone aún de otro nivel de asistencia por ordenador en la forma de líneas impresas, cada una de las cuales cubre un segmento de la ruta planificada, derivada del plan de vuelo del piloto. Estas líneas, con información relativa a la trayectoria de vuelo, la altura, carga útil y tipo de avión, ayudan al controlador a guiar el vuelo a través de su zona de la forma más rápida y más económica.

En el transporte por ferrocarril también está muy difundida la utilización de ordenadores. El trabajo del guardavía ferroviario, si bien no es tan complejo como el de los controladores de tráfico aéreo, tiene varios rasgos en común. Su labor consiste, igualmente, en guiar el tránsito de pasajeros y de mercancías a través de su zona con absoluta seguridad y con el menor costo posible. En Gran Bretaña, la British Rail viene utilizando sistemas de control por ordenador desde mediados de los años setenta, siguiendo el ejemplo que puso en práctica la Southern Pacific Railroad en Estados Unidos. Su TOPS (*Total Operations Processing System*: sistema de procesamiento total de operaciones) cubre todos los aspectos del control de carga, desde llevar el inventario exacto y actualizado al minuto de cada locomotora y de cada componente de material rodante, hasta ensamblar estas unidades para formar trenes completos y determinar sus rutas.

Cada vagón de mercancías posee un número de identificación exclusivo que el ordenador del TOPS graba junto con su localización. Cuando en otro lugar se necesita un vagón de esa clase, éste es asignado a un tren determinado y se apunta su destino. Habiendo cumplido su cometido, el TOPS toma nota de la nueva posición del vagón y queda en condiciones de volver a asignarlo. Si se considera el volumen del tráfico de mercancías que maneja la British Rail (a finales de la década de los setenta había 185 000 vagones distribuidos en casi 2 000 trenes diarios), se comprende la necesidad de utilizar estos sistemas de control por ordenador.

En las estaciones de ferrocarril se presentan varios de los problemas de operación que existen en los aeropuertos, y en ellas se han puesto en práctica muchas soluciones similares. No obstante, en el caso de los ferrocarriles hay que tener en cuenta otro adelanto: la creciente utilización de estaciones sin personal. Se han llevado a cabo experimentos con microordenadores que respondían a las preguntas de los viajeros (y, en algunos casos, anunciaban las salidas y llegadas mediante síntesis de voz). La informática también ha hecho posible la existencia de trenes sin maquinista. En Londres, por ejem-


Roy Ingram

Luz verde y luz roja
Los ingenieros de caminos han venido utilizando durante algún tiempo el ajuste de fase de las señales de tráfico con el fin de regular el flujo de vehículos, especialmente en las calles principales de las ciudades. En la actualidad, los semáforos pueden controlar de manera individual la densidad del tráfico en sus inmediaciones mediante detectores de radar doppler y transferir esta información a un sistema de ordenador. La frecuencia del cambio de luz de los semáforos se puede adecuar, entonces, a las condiciones de cada momento





Cortesía de British Airports Authority

"Se anuncia la salida..."
Quizá sea el tablero de salidas y llegadas el aspecto del sistema informático operativo del aeropuerto que más trasciende al público. Estos dispositivos electromecánicos los actualiza constantemente el ordenador central a medida que va recibiendo nueva información

plo, el metro de la línea Victoria, de la London Transports, dispone de esta capacidad, si bien no se emplea en la práctica debido al recelo que despiertan en el público los trenes sin conductor.

Las representaciones a escala más sofisticadas de redes de ferrocarriles se aproximan mucho a lo que es un sistema de tren sin conductor. Cada locomotora representada posee un código identificador exclusivo, denominado "número de dispositivo", almacenado en un pequeño microordenador basado en un solo chip. El controlador le envía las señales a cada tren modulando la energía conducida a lo largo de las vías, de modo tal que sólo una locomotora será capaz de decodificarla. De este modo, un elevado número de trenes puede circular simultáneamente por el mismo trazado bajo el control directo del microordenador central.

El tren sin conductor es viable porque circula sobre rieles, pero es poco probable que el concepto se utilice para los vehículos de carretera. No obstante, ya se emplean ordenadores en el transporte por carretera, básicamente para el transporte público y las operaciones de carga. En este caso contribuyen a planificar y trazar las rutas de los vehículos, así como a confeccionar horarios (lo cual es una tarea compleja en una gran ciudad, donde es necesario integrar los servicios de autobuses, trenes y metro en un único sistema de transporte público). El problema consiste en mantener en funcionamiento justo la cantidad de vehículos necesarios para que los pasajeros no sufran demoras inaceptables y, al mismo tiempo, no perjudicar la rentabilidad de la empresa. Desde el punto de vista estadístico esto constituye un espinoso problema, pero fue para resolver problemas de esta índole que se desarrollaron por primera vez los ordenadores. Otra tarea que se puede realizar con la aplicación de métodos estadísticos es el trazado de las rutas de los camiones de reparto para minimizar la distancia a cubrir entre cada entrega y la asignación de consignas a cada camión. En este sentido, una variante interesante se observa en el envío de vehículos, especialmente de taxis y de coches policiales, que van "viajando" en vez de regresar a su punto de salida. A la localización de cada vehículo se le da entrada como el nombre de una calle, que el ordenador convierte en la referencia de una cuadrícula. A cada solicitud de envío se le da entrada de la misma forma, y la tarea de relacionar los recursos con las llamadas es una simple cuestión de comparar los dos de acuerdo a reglas predeterminadas.

El sistema denominado *dial-a-bus* ("llame un autobús"), actualmente en funcionamiento en la periferia de Hannover (Alemania), representa uno de los experimentos más interesantes llevados a cabo con ordenadores en el área del transporte público. Basado en una flota de miniautobuses que no siguen ninguna ruta preestablecida, el sistema permite que el usuario telefonee desde la parada del autobús a la estación de control central, indicando su punto de destino. La miniterminal (que tiene el aspecto de una terminal de cajero automático) imprime entonces la hora a la cual llegará el autobús (el margen de tiempo nunca supera los cinco minutos), la duración del viaje y el precio del billete.

El transporte de pasajeros y de mercancías supone el 20 % de todo el comercio internacional. En este campo la utilización del ordenador está más avanzada que en otros y sin duda alguna ha contribuido significativamente a su crecimiento. Si bien la mayoría de los ejemplos que hemos citado se realizan con miniordenadores u ordenadores de unidad principal, muchos de ellos también se pueden efectuar con microordenadores personales.

Carga por ordenador
La industria naviera utiliza los ordenadores en mucha menor medida que otros medios de transporte; una importante área de aplicaciones es la de los servicios de embalaje de carga en contenedores. Los microordenadores juegan un papel muy importante en la "consolidación" (grupo de empresas fletadoras que comparten un contenedor), organización y arreglo de la terminal de contenedores y en la carga del buque contenedor con el fin de asegurar una distribución uniforme del peso



Cortesía de «The Motor Ship»

Hablando idiomas

El BASIC posee una construcción muy familiar y por ello resulta sencillo de aprender, pero no es tan eficaz como otros lenguajes

A menos que su ordenador personal sea un Jupiter Ace (véase p. 150), es casi seguro que el lenguaje de programación que lleva incorporado su aparato es el BASIC. Pero esto no significa que usted esté limitado a esa opción; en efecto, a pesar de que el BASIC está reconocido como un lenguaje particularmente fácil de aprender, hay otros que resultan muchísimo más apropiados para escribir aplicaciones específicas. Para disponer de estos lenguajes en su ordenador le será necesario cambiar las unidades ROM que contienen el intérprete de BASIC, o bien cargar en RAM el nuevo lenguaje; en este último caso, necesitará una máquina que posea una capacidad de memoria razonable, para que quede suficiente RAM libre para contener sus programas. Algunos ordenadores personales, como el Sharp MZ-711, se han anticipado a este problema cargando el intérprete de BASIC también en cassette.



PASCAL-CASTELLANO

CASTELLANO-PASCAL

El PASCAL se desarrolló a principios de la década de los setenta con la intención de convertirlo en el sucesor del BASIC. Su gama de estructuras de datos y de control deriva del grupo de lenguajes FORTRAN/ALGOL, y están pensadas para alentar al estudiante a enfocar la programación de forma sistemática y a escribir códigos bien estructurados y fácilmente comprensibles. Se trata de un lenguaje de programación muy conveniente para desarrollar una buena técnica de programación, aunque esto conlleve que las primeras etapas del aprendizaje de la programación puedan resultarle más difíciles al principiante absoluto, en parte debido a la disciplina que impone el lenguaje y también porque generalmente se compila en vez de interpretarse. No obstante, los programas en PASCAL tienden a ser elegantes, se desarrollan con relativa rapidez y son mucho más fáciles de comprender.

Éste es el equivalente en PASCAL del programa en BASIC:

```
VAR
NAME          :PACKED
               ARRAY (1..30)
               OF CHAR;
AGE, COUNT    :INTEGER;
ANSWER        :PACKED
               ARRAY (1..3)
               OF CHAR;
RUNNING       :BOOLEAN;

BEGIN
RUNNING: = TRUE;
WHILE RUNNING DO
    BEGIN
    WRITE('¿Cuál es su nombre?');
    READLN(NAME);
    WRITE ('¿y cuántos años tiene?');
    READLN(AGE);
    FOR COUNT: = 1 TO AGE DO
        WRITE(COUNT:3,'Hola':10,NAME);
    WRITE('¿Quiere otra pasada?');
    READLN(ANSWER);
    IF ANSWER(1) = 'N'
    THEN RUNNING: = FALSE;
    END;
WRITELN('Adiós', NAME)
END.
```

BASIC-CASTELLANO

CASTELLANO-BASIC

El BASIC, un lenguaje muy difamado, se desarrolló a partir del FORTRAN (que fue uno de los primeros lenguajes de programación de alto nivel y que sigue siendo uno de los lenguajes más populares para aplicaciones científicas y de ingeniería), destinado a los estudiantes universitarios como una forma de introducción a la programación. Dado que fue concebido como método docente, el BASIC por lo general no se compila sino que se interpreta (véase p. 184), esta característica ha sido la causa principal de que se convirtiera en el lenguaje incorporado de casi todos los ordenadores personales. A los lenguajes interpretados no resulta caro ponerlos en práctica, utilizan comparativamente menos memoria del ordenador y son muy adecuados para desarrollar programas.

Ahora el BASIC es un lenguaje eficaz por derecho propio, pero se encuentra limitado por poseer demasiadas versiones no estandarizadas (el BASIC de cada ordenador es exclusivo de esa máquina) y por su carencia de datos especializados y de estructuras de control. A causa de estas desventajas es posible que los programadores autodidactas desarrollen malas técnicas de programación y que una buena técnica de programación pueda ser difícil de aplicar a unos problemas específicos en BASIC.

En este corto programa se pueden apreciar el atractivo y el carácter general de este lenguaje, pero también sus limitaciones:

```
100 INPUT ¿Cuál es su nombre?";N$
200 INPUT "¿y cuántos años tiene?";A
300 FOR K = 1 TO A
400 PRINT K,"Hola  ";N$
500 NEXT K
600 INPUT "¿Quiere otra pasada?";R$
700 IF LEFT$(R$,1) = "S" THEN GOTO 100
800 PRINT "Adiós  ";N$
900 END
```

COMAL-CASTELLANO

CASTELLANO-COMAL

El COMAL se desarrolló para combinar la accesibilidad del BASIC con las eficaces estructuras y el enfoque disciplinado del PASCAL. Por lo tanto se asemeja a ambos y puede que haya sido tomado como modelo para crear el BASIC del BBC Micro, que casi se ha desarrollado como un nuevo lenguaje.

El COMAL ha obtenido un gran éxito en la informática escolar y en los países escandinavos (donde se originó), pero no parece probable que llegue a desplazar a sus dos antecesores como el lenguaje de introducción a la programación.

Éste es el programa "Hola" en COMAL:

```
100  RUNNING: = TRUE
200  WHILE RUNNING DO
300      INPUT"Cuál es su nombre?":N$
400      INPUT"¿y cuántos años tiene?";A
500      REPEAT
600          FOR K: = 1 TO A DO
700              PRINT K,"Hola";N$
800          NEXT K
900      INPUT"Quiere otra pasada";R$
1000     IF R$ = "N" THEN RUNNING: = FALSE
1100 ENDWHILE
1200 PRINT"Adiós";N$
```

LISP-CASTELLANO
CASTELLANO-LISP

El LISP se desarrolló a principios de los años sesenta como un lenguaje para tratamiento de listas y desde entonces se ha venido utilizando ampliamente en el campo de la inteligencia artificial, que implica la continua búsqueda y comparación de listas de datos, relaciones y respuestas. A diferencia del BASIC, que tiende a resaltar la importancia del flujo del programa a través de una secuencia de instrucciones y procedimientos, el LISP es un lenguaje "funcional", en el cual el juego de órdenes elementales se puede construir para conformar funciones más sofisticadas con nombres determinados por el programador. Por ejemplo:

```
(SETQ LISTA1 '(4 7 2 5 1))
```

crea una lista denominada LISTA1 cuyos elementos son los números (4 7 2 5 1).

```
(CAR LISTA1)
```

da el primer elemento de la lista LISTA1 (4, en este caso).

```
(CDR LISTA1)
```

da la lista LISTA1 eliminando el primer elemento:(7 2 5 1), en este caso.

```
(SETQ LISTA1(CDR LISTA1))
```

convertirá la lista de LISTA1 en una copia de sí misma excluyendo su primer elemento.
El LISP también se presta para las aplicaciones "repetitivas": problemas que, después de aplicárseles una función simple, quedan reducidos a un problema menor pero idéntico.

LOGO-CASTELLANO
CASTELLANO-LOGO

El LOGO lo desarrolló un psicólogo que se hallaba trabajando en "inteligencia artificial" mientras impartía clases a sus alumnos. Se asemeja al FORTH tanto en su interactividad como en la utilización de funciones "primitivas" que se pueden incorporar a las funciones definidas por el usuario. Pero encarna el principio fundamental de que la forma de aprender algo consiste en enseñarle a otro (en este caso al ordenador) cómo hacerlo. Se considera un lenguaje innovador que creará una forma completamente nueva de enseñarles a pensar a los niños.
Al LOGO por lo general se lo denomina lenguaje de "tortuga", porque se lo suele utilizar para controlar a un pequeño robot con ruedas denominado "tortuga" (véase p. 34).
El siguiente fragmento en LOGO dibuja una casa simbólica como un cuadrado de dimensiones especificadas con un triángulo encima:

```
TO TRIANGLE LENGTH
   REPEAT 3(FORWARD:LENGTH RIGHT 120)
END
TO SQUARE LENGTH
   REPEAT 4(FORWARD:LENGTH RIGHT 90)
END
TO HOUSE LENGTH
   RIGHT 30
   TRIANGLE:LENGTH
   LEFT 90
   SQUARE:LENGTH
END
```

Ahora al digitar HOUSE 15 se dibujará una "casa" que tendrá una longitud lateral de 15 unidades.

FORTH-CASTELLANO
CASTELLANO-FORTH

El FORTH se parece al LOGO en cuanto que es un lenguaje funcional interactivo, con la importante diferencia de haber sido el primer lenguaje, aparte del BASIC, que se incorporó a un ordenador personal (el Jupiter Ace). El lenguaje consta de varias funciones definidas, denominadas "primitivas", y posee la capacidad de definir funciones nuevas desde el punto de vista de éstas. En FORTH las operaciones matemáticas son "orientadas en pila", lo que significa que a la memoria del ordenador se la trata como una lista de datos que se expande y se contrae, lo que da por resultado que la última operación siempre esté al comienzo de la lista. Otra consecuencia de la "orientación en pila" es que no se utiliza la notación algebraica. En vez de escribir (12 + 4)/2 para hallar la media de 12 y 4, en FORTH usted debería escribir 12 4 + 2/, que es la misma suma que antes expresamos en notación algebraica, expresada ahora en notación polaca invertida.
Todo esto hace que el FORTH sea un tipo de lenguaje muy diferente, que obliga a aplicar un criterio muy distinto para solucionar los problemas y para efectuar los procesos informáticos. En la jerarquía de los lenguajes de alto nivel, se encuentra casi un paso más abajo.
Este fragmento en FORTH define dos palabras nuevas denominadas SHOUT y CHORUS:

```
:SHOUT (imprime "SHAZAM!")
   ."SHAZAM!";
:CHORUS (utiliza SHOUT en un bucle)
   0 DO SHOUT LOOP;
```

Ahora digitar "n" CHORUS hará que SHAZAM! se imprima "n" veces en la pantalla.

En estas páginas le ofrecemos un panorama de los lenguajes de programación más comunes disponibles para ordenadores personales. Tal como ocurre con los idiomas corrientes, mientras más lenguajes de programación domine, más fácil le resultará aprender uno nuevo.

Hacia el firmamento

En astronomía los ordenadores cumplen principalmente dos funciones: almacenar datos de los objetos observados y calcular su posición para facilitar la alineación precisa del telescopio



Marcus Wilson-Smith

Luz guía
La astronomía óptica se simplifica considerablemente cuando se puede formular una predicción exacta de la situación de una determinada estrella o planeta en un día determinado. Un ordenador personal puede datar las localizaciones de los astros y efectuar con rapidez los cálculos necesarios. Con la adición de los motores a impulsos, el ordenador puede incluso determinar la posición del telescopio

Cuando mira el cielo nocturno y observa la estrella más cercana, en realidad la está viendo tal como era hace cuatro años, porque ése es el tiempo que tarda en llegar a nosotros la luz desde Alfa de Centauro. Durante esos cuatro años el microordenador ha pasado de ser una novedad extraña y cara a ser un periférico habitual en los hogares del mundo. La astronomía es una ciencia que utiliza al máximo el potencial del ordenador personal para manipular datos, efectuar cálculos y para la robótica.

Al estudiar el cielo nocturno el astrónomo se enfrenta con tres tipos principales de problema: la observación inicial del objeto celeste, la manipulación de los datos obtenidos a través de la observación y el análisis significativo de esos datos. El ordenador constituye un medio auxiliar muy útil en cada una de estas áreas.

Comencemos por considerar cómo puede ayudar un ordenador personal a efectuar los cálculos necesarios para construir la herramienta básica del astrónomo, el telescopio. La disposición y el diseño de las lentes y los espejos de un telescopio son decisivos para la calidad de la imagen final y se pueden calcular matemáticamente para obtener los mejores resultados posibles. A los astrónomos aficionados a menudo les gusta construir sus propios sistemas ópticos, pero antes de que los ordenadores personales tuvieran una difusión tan amplia, con frecuencia era mucho más rápido y más sencillo armar un diseño experimental que efectuar los laboriosos cálculos necesarios para los diagramas de rayos de luz. Con un ordenador ahora se pueden realizar en unos pocos minutos cálculos que antes podrían haber tardado una semana.

También el ordenador nos puede ser de gran ayuda para localizar una estrella en el firmamento. Las estrellas no son objetos fijos: siguen trayectorias a través del cielo en el curso de la noche (efecto provocado por la rotación de la Tierra) y sus posiciones en el cielo están sujetas a las variaciones de las estaciones. El procedimiento que se sigue para localizar una estrella es similar al sistema de latitud y longitud que se emplea en la geografía terrestre. Si imaginamos un sistema de coordenadas proyectadas a través de la superficie interior del cielo nocturno, cada objeto celeste se puede localizar mediante dos coordenadas denominadas *declinación* y *ascensión recta*.

Todos los objetos se pueden señalar después en un mapa estelar de acuerdo a las coordenadas de cada uno, y se pueden combinar los diversos mapas para elaborar un atlas del firmamento. Este tipo de atlas permite observar los planetas y otros objetos que se mueven en contraste con el fondo de las estrellas fijas, o bien descubrir objetos celestes completamente nuevos, como los cometas. Estos atlas estelares en la actualidad están siendo incluidos en bases de datos para ordenadores personales. Estos programas de bases de datos también incluyen información relativa a la brillantez o luminosidad de cada uno de los cuerpos celestes, la calidad espectral de la luz que emiten (obtenida cuando se la hace pasar a través de un prisma o se la analiza con un espectrómetro) y el tipo y edad de la estrella. Toda esta información se puede visualizar en el televisor o el monitor del ordenador, en forma de mapas que muestran el cielo desde cualquier latitud en el momento que se determine.

Debido a la rotación de la Tierra, las estrellas desaparecen de la vista en unos pocos minutos, incluso para los telescopios de gran campo visual. Si

se utiliza un telescopio que pueda enfocar sólo un área estrecha del cielo, resulta esencial desviar el telescopio de manera continua para compensar el movimiento de la Tierra. Durante muchos años se han empleado motores de acción mecánica, pero recientemente el aficionado ha tenido acceso a los sistemas controlados por ordenador. El telescopio está montado sobre un eje "ecuatorial" que señala el norte verdadero, y el motor hace rotar el eje exactamente a la velocidad a la que se mueve la Tierra, con lo que se consigue mantener al objeto dentro del campo visual. Acoplados al eje y al telescopio hay codificadores y digitalizadores axiales, que le envían al ordenador una señal digital para las coordenadas celestes y controlan la actividad del motor de actuación. De esta forma, se puede dejar que el telescopio, bajo el control del ordenador, siga durante toda la noche la pista de una estrella tenue, mientras su imagen se va componiendo lentamente en una placa fotográfica. En Estados Unidos existe un adaptador llamado Celestial Navigator Mk II, que conecta un telescopio con el ordenador personal a través de interfaces de ampliación.

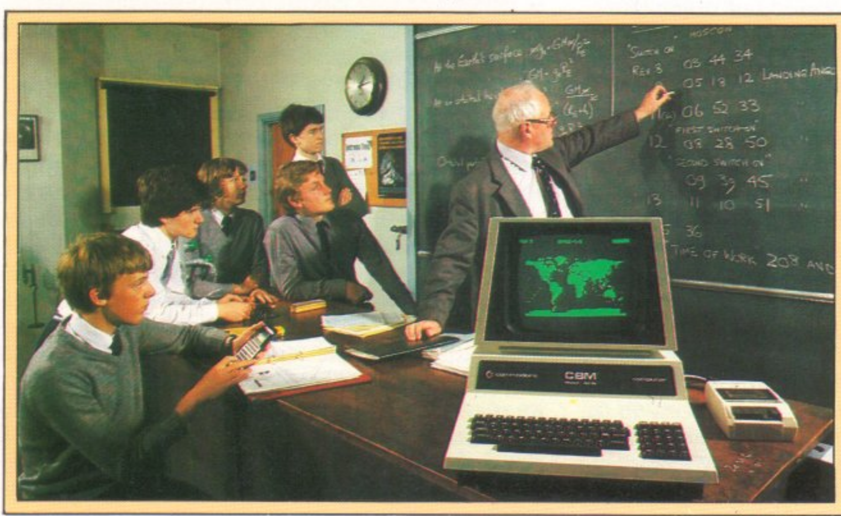
En ese mismo país, algunos astrónomos aficionados han llegado incluso a adaptar las más recientes tecnologías de reconocimiento y de síntesis de voz al campo de la astronomía. Al ordenador se lo programa para que reconozca ciertas palabras-órdenes; de este modo, si un observador penetra en el edificio y dice "abrir", la cúpula se desliza y se abre; con una ulterior orden ("hacer rotar la cúpula") se consigue que los motores la hagan girar sobre sus ruedas. La síntesis de voz también es muy útil en la oscuridad de un observatorio para hacer que el ordenador produzca una salida de información. Podría, por ejemplo, contar en voz alta las señales de tiempo para ayudar al observador a tomar exposiciones fotográficas exactas.

Los astrónomos profesionales utilizan telescopios que registran parte del espectro electromagnético además de la luz, como ondas de radio o rayos X. Con el aumento de tamaño de la apertura del telescopio y el consiguiente incremento en el peso, los problemas de ingeniería se volvieron muy críticos. En 1964, el Jodrell Bank Mark II, disco elíptico de 38 por 25 metros situado cerca de Manchester, se constituyó en el primer telescopio que utilizó un ordenador digital para convertir las coordenadas celestes en las instrucciones que necesitaban los diversos motores de actuación; el proceso se efectuaba continuamente cuatro veces por segundo.

Los ordenadores también ayudan al telescopio al tener en cuenta aquellos cambios atmosféricos que, como las variaciones de temperatura y humedad, refractan y desvían la luz mientras ésta atraviesa la atmósfera. Asimismo, pueden compensar y corregir las distorsiones de la imagen recibida a través del telescopio, en virtud de varias técnicas destinadas a mejorar la imagen.

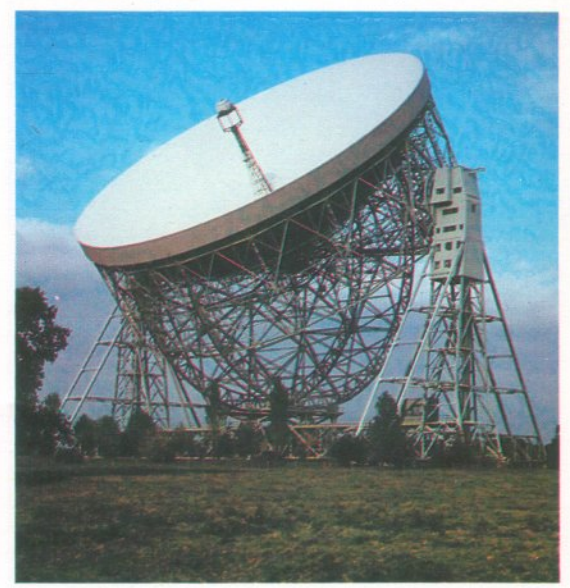
La astronomía no sólo se sirvió de los ordenadores sino que también ha contribuido al desarrollo de la ciencia informática. El lenguaje FORTH lo creó un astrónomo, Charles H. Moore, en 1971 en el observatorio de Kitt Peak (Arizona) para controlar los radiotelescopios y procesar datos.

En la actualidad están apareciendo libros sobre programación personal destinados a los entusiastas de la astronomía e incluso se edita una revista, *Apex*, en la cual los usuarios de ordenadores personales publican e intercambian programas. Se han escrito programas para calcular las fechas de la Pascua de Resurrección, la conversión de las fechas históricas al calendario juliano, la salida y la puesta de la luna y tablas diarias para determinar el sitio exacto del firmamento donde los astrónomos esperan ver el regreso del cometa Halley en 1986.



Los mejores, los de una escuela de humanidades
Los alumnos y el personal docente de la Kettering Grammar School, de Northamptonshire (Gran Bretaña), gozan desde hace tiempo de gran estima por el trabajo que realizan en el campo de la astronomía, especialmente en lo que se refiere al seguimiento de satélites. En varias ocasiones, la KGS ha sido la primera estación de observación en detectar la presencia de un nuevo satélite en órbita

John Drisdale/Colorific



Estrellas con radio
A raíz del descubrimiento de la presencia de objetos de galaxias remotas que emiten radio, se construyó este gigantesco radiotelescopio en Jodrell Bank, al suroeste de Manchester. Los radiotelescopios, que operan a modo de inmensas antenas, pueden detectar objetos invisibles hasta para los telescopios ópticos más potentes

Paso a paso

Cuando hay que medir el movimiento lineal o angular por medio de un sensor óptico, la codificación binaria no es fiable: por ello se inventó el código de Gray

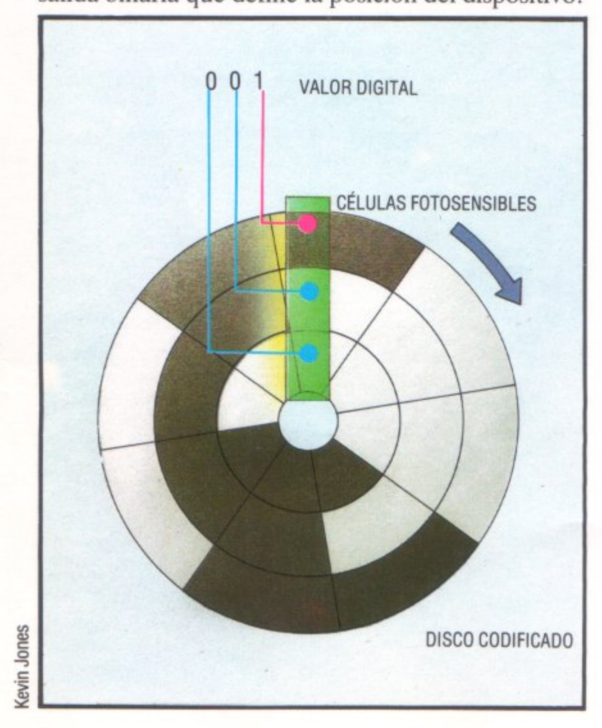
Extraños fuera
Esta tabla muestra los equivalentes en código de Gray de los decimales del 0 al 15.

Decimal	Binario	Cód. Gray
0	0000	0000
1	0001	0001
1	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

Hay muchas tareas en las que se ha de determinar y transmitir con gran precisión al ordenador la posición física de un objeto móvil. En robótica, por ejemplo, el ordenador ha de estar enterado de las posiciones y las orientaciones de todos sus miembros; y en la manipulación de herramientas por máquinas controladas por ordenador, se debe determinar exactamente la posición de la mesa fresadora. ¿Cómo se puede convertir una posición en un valor binario para ser procesado por ordenador?

Uno de los métodos consiste en utilizar un sistema analógico. Éste puede implicar una conexión del dispositivo móvil a una resistencia variable y el paso del voltaje resultante a un conversor de analógico a digital (o directamente a la puerta analógica, en caso de que su ordenador poseyera una). Sin embargo, este sistema no es muy preciso y los componentes mecánicos del mismo son susceptibles de gastarse y romperse.

La alternativa consiste en imprimir un código binario en el dispositivo móvil y leérselo directamente al ordenador. El código generalmente se imprime como un patrón de bloques negros y blancos a lo largo de la parte superior del dispositivo y se lee mediante una fuente luminosa que brilla en un patrón junto con una línea de células fotosensibles, cada una de las cuales es responsable de un dígito del patrón binario. A medida que se mueve el objeto sobre el que se está trabajando, el patrón situado debajo de las células luminosas cambia y da una salida binaria que define la posición del dispositivo.



Ángulo visual
La posición angular de un dispositivo se le puede leer al ordenador por medio de un disco que lleva impreso un código. En el dispositivo hay una luz que brilla y una línea de células fotosensibles detecta el patrón. Se produce una señal digital en paralelo que cambia a medida que se mueve el dispositivo. El problema de utilizar como código el binario, es que si el disco se detiene en la juntura entre dos valores se puede producir un resultado sin ningún significado. El código de Gray da solución a este problema

Además de las disposiciones lineales, también se utilizan patrones radiales para codificar el movimiento angular, como el de la articulación de un brazo robot.

Surgen problemas, sin embargo, cuando el dispositivo se mueve desde un código binario a otro y, especialmente, si tiene que descansar a medio camino entre ambos. La precisión de la impresión tiene una tolerancia finita, y cuando el dispositivo se detiene en una juntura entre dos códigos, las células luminosas podrían optar por leer cualquiera de los dos. Si la pieza de trabajo llegara al punto de descanso con las células luminosas brillando en la juntura entre la posición binaria 11 (1011) y 12 (1100), por ejemplo, entonces sólo se podría confiar en el bit más significativo (es decir, el situado más hacia la izquierda), para convenir la salida correcta, mientras que la lectura de los valores de las otras tres células luminosas sería conflictiva. En algunas situaciones cambian todos los bits, como en la juntura entre el siete (0111) y el ocho (1000) binarios, de manera que hasta la más insignificante imprecisión de impresión produciría lecturas incorrectas en todas las células. El resultado podría ser un valor totalmente falso para la posición y el ordenador no tendría posibilidad alguna de saber que esto había sucedido. Las consecuencias podrían ser desastrosas.

Lo que se necesita, por consiguiente, es un sistema de cuenta alternativo al binario, en el que a cada incremento sólo se modifique un bit. Esto significaría que en una juntura cualquiera sólo un bit podría ser dudoso, y que la salida sería errónea en no más de una posición. Este código alternativo se denomina código de Gray y está determinado exclusivamente por estos requerimientos: moviéndose de un valor al valor siguiente, cambia sólo un único bit y éste habría de ser el bit situado más hacia la derecha, lo cual daría un único patrón. De manera que si empezamos con 0000, como en binario, el número uno se representará por 0001. Sin embargo, para representar dos, debemos cambiar el segundo bit para obtener 0011. Para tres, ahora se puede cambiar el primer bit, obteniendo 0010. Observe la diferencia con la secuencia binaria para los mismos números: 0000, 0001, 0010, 0011.

El panel muestra este proceso ampliado hasta el equivalente del decimal 15, junto con los equivalentes en binario a modo de referencia. Como ejercicio, usted puede calcular los valores superiores en código de Gray.

Los ordenadores se podrían diseñar para que internamente efectuaran la aritmética y las funciones en código de Gray, pero ello sería ineficaz e innecesario. Por lo tanto, se necesita algún medio para convertir de Gray a binario, que se pueda llevar a cabo en hardware o en software.

Apple IIe

Muchos se preguntan la razón de su prolongada vigencia; se debe a que el Apple II es aún uno de los microordenadores más versátiles y, además, dispone de más accesorios que ningún otro

En muchos sentidos, el Apple II puede ser considerado como un pionero; si bien no fue el primer microordenador que apareció en el mercado, fue a raíz de su comercialización que capacidades como color, gráficos de alta resolución y sonido incorporado se hicieron accesibles al usuario de ordenadores personales. Sin embargo, aunque estas configuraciones del Apple son importantes, las realmente significativas, que le permitieron (y le permiten) alcanzar tan extraordinarias cotas de popularidad, resultan mucho menos patentes a primera vista.

La cualidad más manifiesta fue el alto nivel de su documentación, elaborada con el propósito de que los usuarios se hicieran una idea lo más clara posible de todos los aspectos de la máquina. Este hecho contradecía abiertamente la actitud de algunas empresas informáticas de entonces, que mantenían (y en algunos casos siguen manteniendo) un silencio estricto acerca del interior de sus productos. Como resultado directo de esta información de tan fácil acceso, se difundió otra importante configuración del Apple II.

Esta segunda capacidad era la fila de "conectores de ampliación" que ocupa la sección trasera del tablero de la máquina. Es precisamente la adaptabilidad de estas ranuras lo que ha hecho posible la amplia gama de accesorios para la máquina que se encuentran en el mercado. A su vez, esta diversidad le ha proporcionado al Apple una versatilidad extraordinaria.

Muchos ordenadores, por supuesto, poseen alguna forma de conector para ampliación (por lo general uno o dos), pero a menudo con una zona de memoria bastante pequeña. El Apple posee siete en la última versión (el IIe) y ocho en los modelos anteriores (II y II+), cada uno de los cuales aparece para la CPU como dos secciones de memoria (una bastante pequeña, y la otra, muy útil, de 2 Kbytes).

Por consiguiente, una ficha para periféricos que se enchufe en el Apple podrá tener 2 048 bytes para controlar el programa. Ello hace que utilizar una ficha de este tipo resulte muy sencillo, puesto que no es necesario conectar programas activadores especiales ni reescribir el sistema operativo.

Actualmente existe una extensa selección de fichas para el Apple II, que van desde interfaces para input/output relativamente sencillas hasta unidades muy avanzadas, como lápices ópticos y grandes fichas de RAM que pueden ampliar la memoria a un Megabyte o más. No obstante, dado que el 6502 puede direccionar sólo 64 Kbytes, la memoria está dispuesta en "bancos" de 64 Kbytes, que se seleccionan uno cada vez. Incluso se pueden enchufar ordenadores con avanzados chips de CPU de 16 o 32 bits, para funcionar en paralelo con el procesador 6502 de la máquina, con lo que se puede crear un sistema que posea un potencial igual (o superior) al de un miniordenador.

Naturalmente, se ha desarrollado una amplísima gama de software para aprovechar esta riqueza de hardware, y en la actualidad la biblioteca de programas del Apple debe de ser, con toda probabilidad, la más grande del mundo, mayor incluso que la lista de programas que utilizan el sistema operativo estándar CP/M. De hecho, la biblioteca de CP/M se puede incluir en la lista del Apple, porque aunque este ordenador posea un procesador 6502, puede ejecutar programas CP/M con la ayuda de uno de los accesorios más populares: una ficha Z80. (El CP/M funciona solamente con un microprocesador Z80.) Una vez enchufada en una ranura del Apple, la máquina funcionará como un ordenador CP/M normal.

Aunque bastante pequeño y de aspecto modesto, el Apple II es una máquina que suele ser objeto de superlativos. Es el ordenador personal que posee mayor número de sistemas operativos diferentes (11), de lenguajes (al menos 27) y editores de texto (una docena o más) que ninguna otra máquina.

La historia de este ordenador está lejos de haber terminado; en el primer semestre de este año se lanzará al mercado un sistema operativo para la máquina completamente nuevo. Se llamará ProDOS y posee muchas configuraciones que harán que el Apple II supere en rendimiento a algunos sistemas más caros.

El teclado del Apple
Ha sido diseñado de conformidad con los estándares más elevados, teniendo presente el tratamiento de textos. Las teclas están adecuadamente esculpidas y el corte transversal forma un arco para comodidad de uso. Las dos teclas marcadas con el logotipo de Apple a un lado de la barra espaciadora son teclas de control que se utilizan en programas aplicativos. A primera vista, la tecla RESET (restablecer) podría parecer que está demasiado cercana a la tecla DELETE (borrar). No obstante, para restaurar el ordenador se debe mantener pulsada CTRL y después apretar RESET

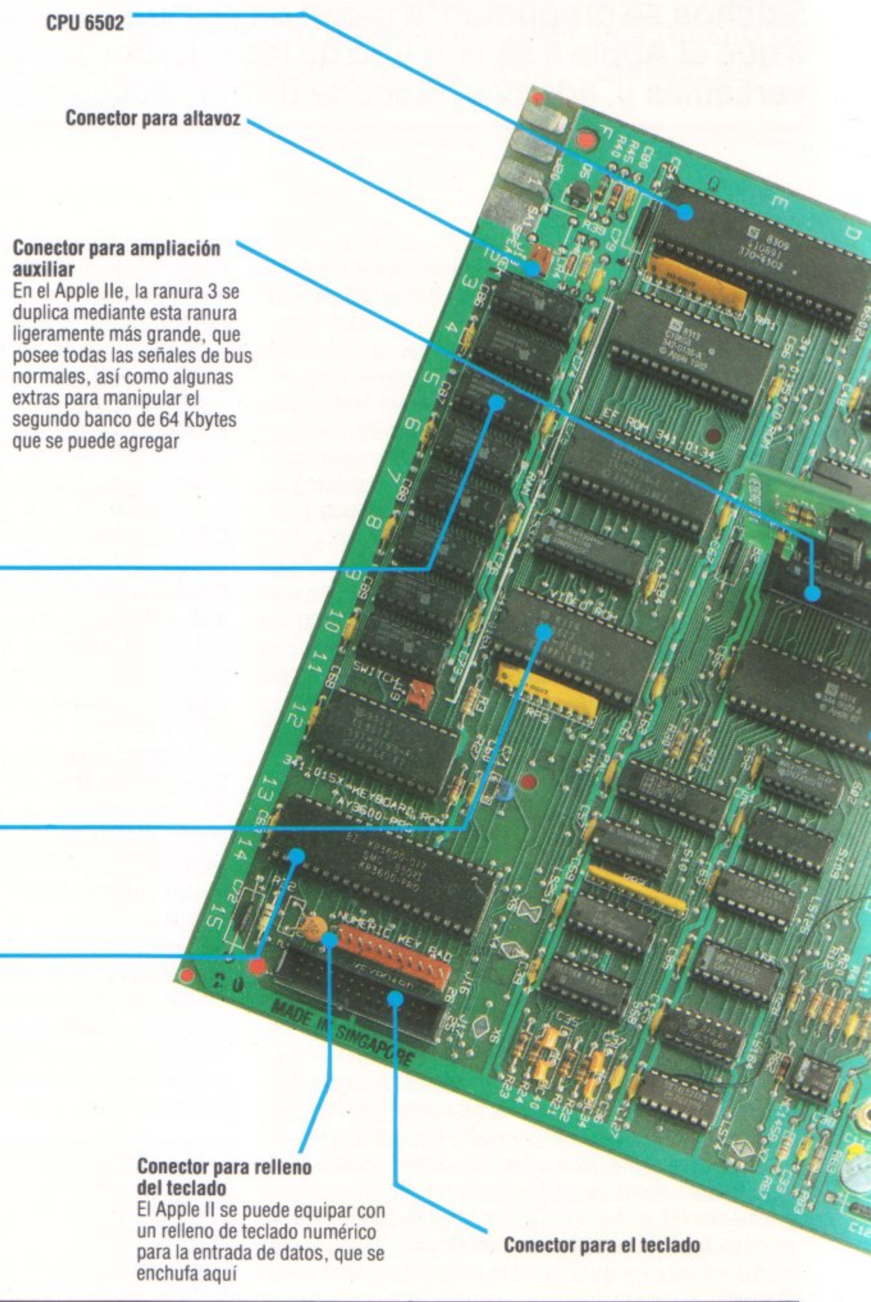


Chris Stevens



Monitor
El Apple IIe funciona con cualquier monitor especializado, pero la propia unidad de Apple es la que mejor armoniza con su diseño físico. Puede visualizar 80 columnas de texto y está equipado con un filtro antideslumbrante para reducir la fatiga visual

Chris Stevens



CPU 6502

Conector para altavoz

Conector para ampliación auxiliar
En el Apple IIe, la ranura 3 se duplica mediante esta ranura ligeramente más grande, que posee todas las señales de bus normales, así como algunas extras para manipular el segundo banco de 64 Kbytes que se puede agregar

RAM
El Apple IIe posee una RAM de 64 Kbytes lineales, retenidos en ocho chips 4164, pero el trazado de mapas es bastante más complejo, porque las direcciones de los chips para periféricos están situadas en el medio de la RAM

ROM de video
Los caracteres que se imprimen en la pantalla los genera esta ROM, que existe en diversos juegos de lenguajes diferentes

ROM del teclado
El trazado del mapa del teclado está controlado por esta ROM, que se sustituye para satisfacer las necesidades de los diferentes países; Francia, por ejemplo, puede tener teclados AZERTY

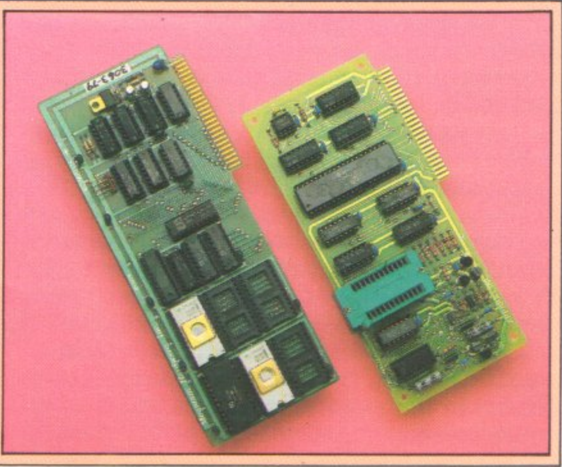
Conector para relleno del teclado
El Apple II se puede equipar con un relleno de teclado numérico para la entrada de datos, que se enchufa aquí

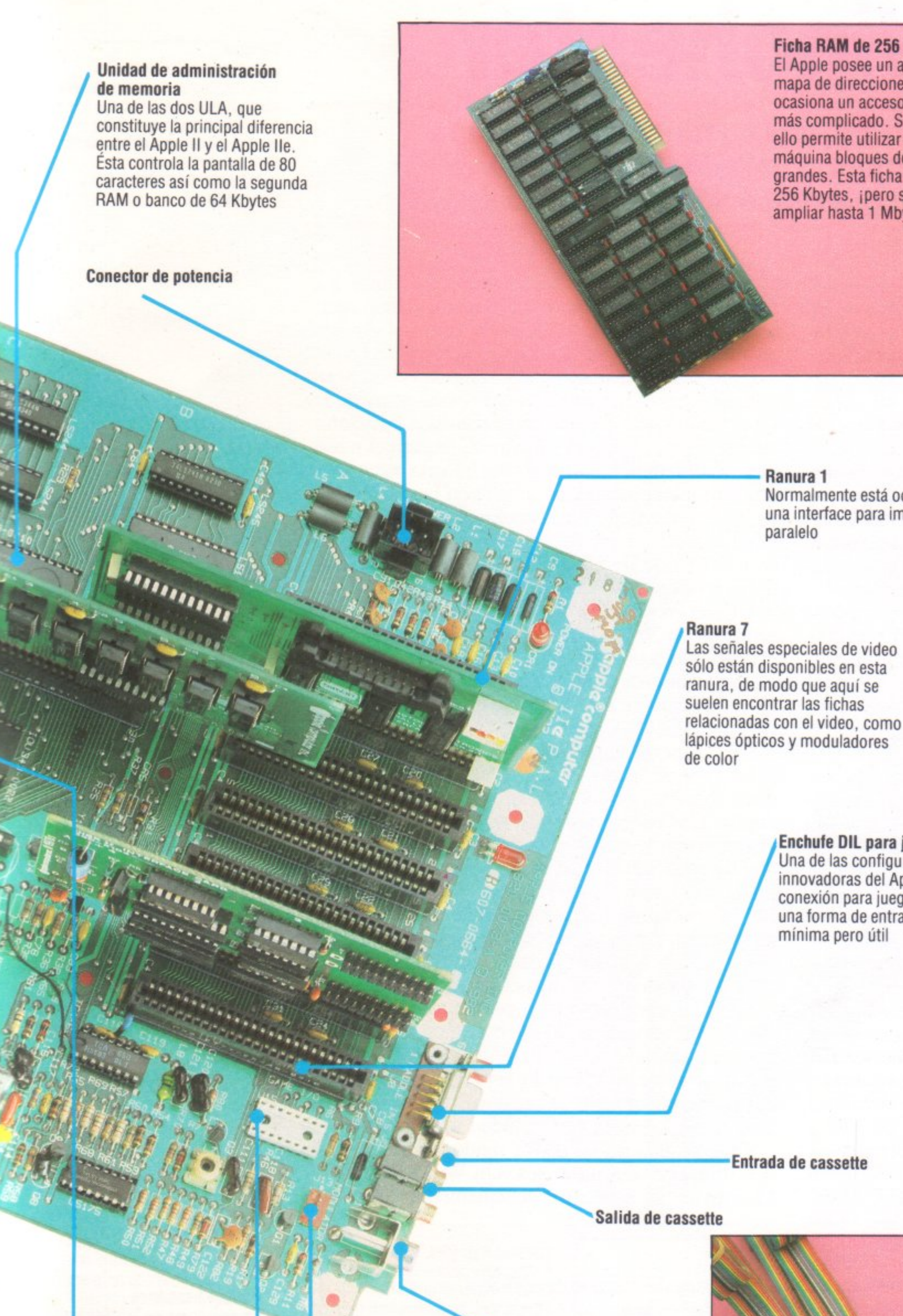
Conector para el teclado



Unidad de disco
Los discos de Apple no son inteligentes, de modo que se debe acoplar una ficha controladora en el tablero principal del ordenador. Hoy, los 143 Kbytes de capacidad son muy pocos, pero son suficientes para muchas aplicaciones

Blower EPROM y Ficha ROM
La amplia gama de herramientas de desarrollo disponibles para el Apple lo hacen ideal para utilizarlo como un sistema de desarrollo para nuevos programas. El blower EPROM se puede emplear para producir unidades EPROM, que después se pueden insertar en una ficha ROM. Cuando se haya perfeccionado el programa, se lo colocará en forma de ROM si es que se va a vender en gran número





Unidad de administración de memoria
Una de las dos ULA, que constituye la principal diferencia entre el Apple II y el Apple IIe. Ésta controla la pantalla de 80 caracteres así como la segunda RAM o banco de 64 Kbytes

Conector de potencia

Ficha RAM de 256 Kbytes
El Apple posee un avanzado mapa de direcciones, lo que ocasiona un acceso ligeramente más complicado. Sin embargo, ello permite utilizar en la máquina bloques de RAM muy grandes. Esta ficha transporta 256 Kbytes, ¡pero se puede ampliar hasta 1 Mbyte!

Ranura 1
Normalmente está ocupada por una interface para impresora en paralelo

Ranura 7
Las señales especiales de video sólo están disponibles en esta ranura, de modo que aquí se suelen encontrar las fichas relacionadas con el video, como lápices ópticos y moduladores de color

Enchufe DIL para juegos
Una de las configuraciones más innovadoras del Apple fue la conexión para juegos, que daba una forma de entrada analógica mínima pero útil

Entrada de cassette

Salida de cassette

Salida de video compuesto

Conector auxiliar para video

Unidad de input/output
La ULA que manipula el direccionamiento del conector auxiliar

Enchufe-D para juegos
El enchufe normal de 16 patillas para conexión de juegos es demasiado frágil para utilizarlo a diario, por lo cual el Apple IIe está equipado con un pequeño enchufe-D en paralelo

APPLE IIe

DIMENSIONES
460 × 385 × 115 mm

VELOCIDAD DEL RELOJ
1 MHz

MEMORIA
ROM de 16 Kbytes
RAM de 64 Kbytes
Ampliables a 128 Kbytes o más

VISUALIZACION EN VIDEO
24 líneas de 40 caracteres, sólo monocromática. Gráficos de baja resolución de 48 × 40 en 16 colores. Gráficos de alta resolución de 192 × 280 en 6 colores

INTERFACES
Cassette, video compuesto, 7 ranuras de ampliación, conexión para juegos

LENGUAJE SUMINISTRADO
BASIC Applesoft

OTROS LENGUAJES DISPONIBLES
La mayoría de los lenguajes alternativos comunes, así como algunos menos conocidos

VIENE CON
Manuales de instalación y de BASIC, cable para TV

TECLADO
62 teclas de gran calidad

DOCUMENTACION
La documentación que acompaña a la máquina es de un nivel muy elevado, si bien el material avanzado necesario para una comprensión absoluta de la máquina se ha de adquirir por separado y es bastante caro. Existe gran número de libros que satisfacen todos los niveles de interés y, en este sentido, probablemente se trate de la máquina que ha sido estudiada con mayor profundidad

Ficha de I/O para fines generales
En algunas ocasiones una ficha puede tener tantas posibles funciones que una ROM-controladora limitaría al usuario y se convertiría en un riesgo. Esta ficha, que posee dos versátiles adaptadores para interface 6522, constituye un ejemplo de ello. Posee 40 líneas de I/O controlables independientemente, dos registros de cambio que se utilizan para convertir datos de modalidad en paralelo a en serie, y cuatro cronometradores de 16 bits

Chris Stevens



Discos Winchester

Estos discos rígidos requieren para su funcionamiento unas condiciones de máxima limpieza. Sellados en el interior de su carcasa, proporcionan una gran capacidad y un rápido acceso

Aunque en el transcurso de los últimos años los ordenadores personales han asimilado muchas de las configuraciones de las pequeñas máquinas de gestión, hay un área de la tecnología de los ordenadores en la que carecen relativamente de sofisticación: el espacio del almacenamiento en disco. Mientras que el usuario de un ordenador personal estaría muy satisfecho de tener un único disco flexible capaz de retener 100 Kbytes de datos, los requerimientos de las máquinas contables exigen un espacio considerablemente mayor. Una pila de discos flexibles, con toda la información empresarial diseminada entre ellos, no constituye ninguna solución para esta gran necesidad de almacenamiento; y, por consiguiente, se desarrollaron discos rígidos, capaces de almacenar cantidades mucho mayores de datos. La IBM, en la década de los sesenta, inició los primeros trabajos para crear estos discos rígidos. Dado que los discos originales poseían una capacidad de almacenamiento de 30 Megabytes en cada unidad, se los apodó "30/30" y a partir de allí, por analogía con el rifle, discos "Winchester".

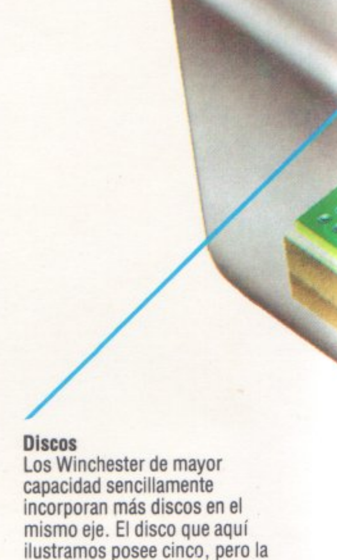
Los Winchester utilizan discos rígidos en lugar de los flexibles de plástico empleados en los discos flexibles. Ello permite incrementar el número de pistas del disco desde un máximo operativo típico de 96 pistas por pulgada de los discos flexibles a varios centenares en los Winchester. Con la creciente sofisticación de la tecnología del disco se ha hecho algo común disponer de cinco, 10 o incluso 20 Megabytes de almacenamiento en una caja de las mismas dimensiones que una unidad de disco flexible de 5 ¼ pulgadas.

Ahora los usuarios de ordenadores personales están comenzando a beneficiarse de la tendencia a emplear los discos rígidos gracias a la disponibilidad en el mercado de los microfloppies Sony de 3 ½" y los Hitachi de 3". Éstos son discos semirrígidos y pueden almacenar tanta información como cualquier disco flexible de 5 ¼ pulgadas. El BBC Micro y el Oric-1 son dos de las primeras máquinas que dispusieron de este tipo de dispositivos como accesorios, mientras que ordenadores como el Apricot de ACT los incluyen ya como accesorios estándar.

Este incremento de la densidad de almacenamiento ha creado, sin embargo, otros problemas. La precisión que exige el mecanismo de posicionamiento de la cabeza, por ejemplo, requería un método completamente nuevo para hacerla mover. La solución a este problema se halló en la industria de audio: con frecuencia las cabezas Winchester se colocan en posición mediante una bobina electromagnética del tipo de las que llevan los altavoces. Al hacer pasar una corriente eléctrica a través de una bobina se genera un campo magnético que a su vez hace que un "brazo" de hierro maleable situado en el centro de la bobina se mueva una distancia precisa. Acoplando la cabeza al extremo de este "brazo" (por supuesto, convenientemente aislada de todo efecto magnético), se consigue que ésta se desplace sobre la superficie del disco con rapidez y precisión.

La cabeza "vuela" a través de la superficie del disco sobre un colchón de aire, sin llegar a tocarla nunca. Ello reduce significativamente el desgaste y el deterioro de los discos, pero implica que éstos han de estar sellados en cajas herméticas para evitar el contacto del polvo y otros cuerpos extraños. Por lo general, se hallan fijos dentro de la unidad y no se pueden retirar, si bien en la actualidad están empezando a aparecer discos Winchester con cartuchos separables. Estos cartuchos suelen ser autosoldables y se abren sólo para permitir que la cabeza acceda a la superficie del disco cuando se inserta el cartucho en la unidad. Para evitar que entre el polvo, se suele mantener la presión del aire dentro del cartucho por sobre la presión exterior, y todo el aire que se bombea en la unidad se filtra antes.

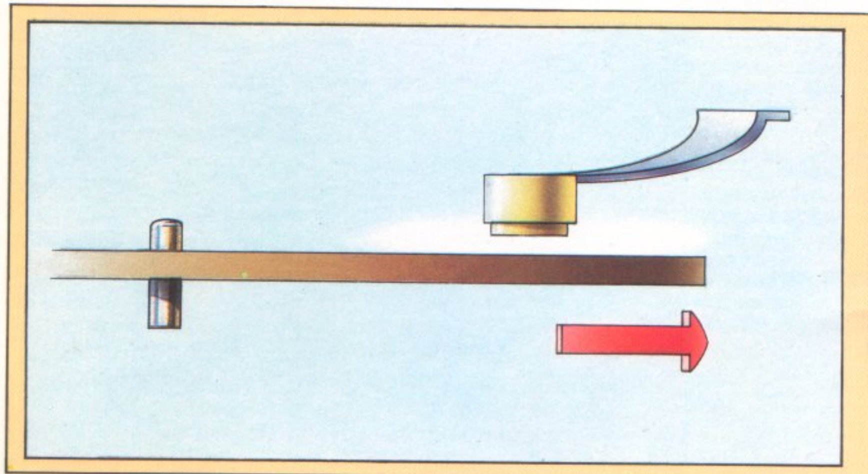
Otra ventaja de los discos rígidos es que se pueden utilizar discos múltiples. Un Winchester de 10 Megabytes se construye sencillamente colocando



Carcasa
La mayoría de las unidades Winchester vienen dentro de una carcasa de aleación colada, a la cual deben gran parte de su peso. Esta carcasa es necesaria para mantener a los componentes alineados con total precisión

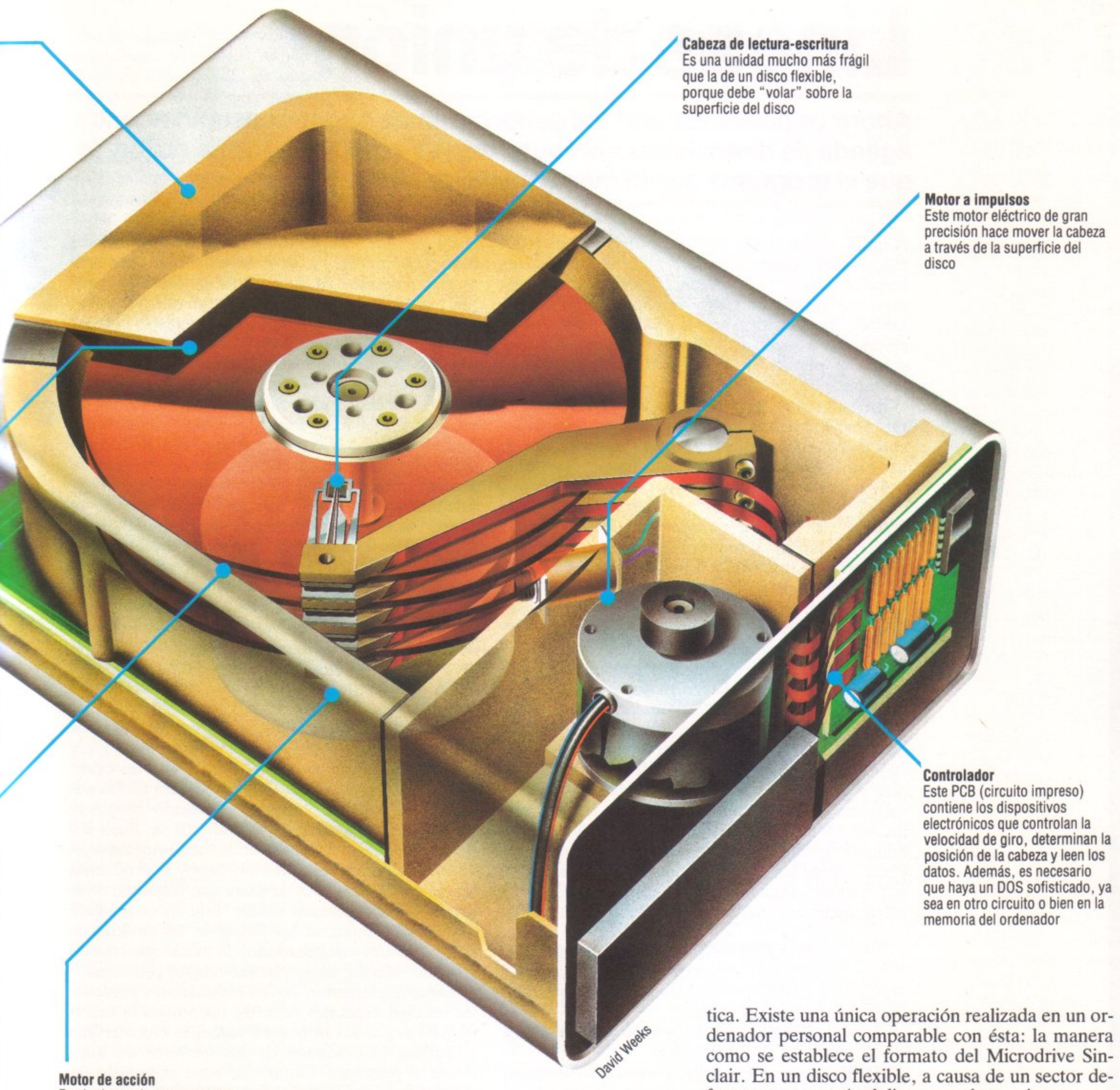
Discos
Los Winchester de mayor capacidad sencillamente incorporan más discos en el mismo eje. El disco que aquí ilustramos posee cinco, pero la mayoría poseen dos o tres. Las cabezas de lectura-escritura están conectadas entre sí, de modo que sólo se puede leer un bloque cada vez

Sellado hermético
La maquinaria de la unidad está herméticamente sellada para impedir que partículas de polvo o humo, provenientes del exterior, "choquen" contra la cabeza



Cabezas flotantes
A diferencia de la unidad de disco flexible, en la cual la cabeza hace contacto con la superficie de grabación, en una unidad de disco Winchester la cabeza "flota" muy cerca de la superficie. El disco gira tan rápidamente que el "efecto Kelvin" crea un colchón de aire sobre el cual se apoya la cabeza. Si ésta "aterrizara" contra el disco, probablemente eliminaría la superficie de grabación magnética

Kevin Jones



Cabeza de lectura-escritura
Es una unidad mucho más frágil que la de un disco flexible, porque debe "volar" sobre la superficie del disco

Motor a impulsos
Este motor eléctrico de gran precisión hace mover la cabeza a través de la superficie del disco

Controlador
Este PCB (circuito impreso) contiene los dispositivos electrónicos que controlan la velocidad de giro, determinan la posición de la cabeza y leen los datos. Además, es necesario que haya un DOS sofisticado, ya sea en otro circuito o bien en la memoria del ordenador

Motor de acción
En el mismo eje, que a su vez hace girar el disco, hay montados un motor de CD (corriente directa) y un pequeño generador. La salida del generador es una medida de la velocidad del motor y se alimenta de un circuito de control especial. Así es como se logra determinar con tanta precisión la velocidad

en la misma caja dos discos de cinco Megabytes. Se consigue administrar todo este espacio de almacenamiento dividiéndolo en una gran cantidad de secciones. Para mantener la compatibilidad con el software existente, que en el momento de ser diseñado estaba previsto que funcionara sólo con discos flexibles, estas secciones generalmente se aproximan a la capacidad de un floppy normal. Un disco Winchester se parece, al menos para el ordenador, a gran cantidad de unidades de disco flexibles separadas.

Cuando se le da el formato a un disco Winchester (es decir, cuando se marcan por primera vez las pistas y sectores), el DOS (véase p. 324) ha de ser capaz de saltarse los "sectores malos", los que posean defectos en la superficie de grabación magnética. Existe una única operación realizada en un ordenador personal comparable con ésta: la manera como se establece el formato del Microdrive Sinclair. En un disco flexible, a causa de un sector defectuoso, se rayaría el disco completo; mientras que en un Winchester, el programa para formato tan sólo apunta que hay un sector determinado inutilizable e impide su ulterior uso. Al fin y al cabo, si se dispone de cinco millones de bytes de espacio, ¿quién puede echar de menos unos pocos centenares?

Siguiendo las tendencias marcadas por los avances que se están produciendo en las otras áreas de la industria de los ordenadores, el disco Winchester se está volviendo más pequeño en tamaño, y ahora existe un microdisco Winchester de 5 Megabytes. Con el acelerado perfeccionamiento que están experimentando las unidades de almacenamiento en disco, la posibilidad de que un ordenador personal corriente configure un almacenamiento de datos de 10 Megabytes no está demasiado lejana.

Líneas de unión

Ahora ya podemos unir los subprogramas que procesarán nuestra agenda de direcciones computerizada y analizar la forma de lograr que el programa sea lo más sencillo posible

Aunque aún debamos dar forma final a muchos detalles del programa para la agenda de direcciones computerizada, la estructura global ya debería estar bastante clara. Llegados a este punto del desarrollo de un programa de cualquier dimensión, resulta útil dibujar un diagrama de bloques del programa y pensar en el flujo de actividades del mismo.

En este punto, el programador debería considerar también los aspectos del programa relacionados con la "interface humana" y la "imagen para el usuario". Pocos fabricantes de software dedican la debida atención a estos dos importantes conceptos.

La "interface humana", definida en términos sencillos, es la "ergonómica" del software, o el nivel de sencillez de su utilización. La "imagen para el usuario" se refiere a la forma en que éste percibe el programa que utiliza. Vamos a analizar estos conceptos en relación a nuestro programa tal como lo hemos desarrollado hasta el momento y estableceremos hasta qué punto podremos aplicarlos.

La tabla muestra los principales bloques del programa que hemos realizado hasta ahora. Convencionalmente, hemos utilizado nombres de seis caracteres para los procedimientos o subrutinas, de siete caracteres (incluyendo $) para las matrices, de cuatro caracteres para las variables numéricas simples y de cinco caracteres (incluyendo $) para las variables alfanuméricas simples. Para las variables locales (dentro de bucles, p. ej.) por lo general hemos empleado una única letra.
Cada uno de los grandes bloques del programa de la segunda columna necesita volver a dividirse en subunidades, y éstas se habrán de volver a refinar hasta que obtengamos todos los detalles suficientes para escribir el código verdadero en BASIC. Los procesos implicados en esta forma de "refinamiento por pasos" ya los hemos ilustrado en muchos de los bloques en capítulos anteriores de nuestro curso.

BLOQUES PRINCIPALES DEL PROGRAMA

	CREMAT	(crea matrices e inicia variables)
INICIL	LEARCH	(lee los archivos desde cinta o disco)
	ESBAND	(establece banderas y modifica variables)
PRESEN		(imprime mensaje presentación)
ELECCN	IMMENU	(imprime menú de opciones)
	ASOPCN	(asigna opción a OPCN)
	ENCREG	(localiza e imprime un registro)
	ENCNOM	(localiza nombres a partir de entradas parciales)
	ENCIUD	(localiza registros de una ciudad específica)
	ENCINI	(localiza nombres a partir de iniciales)
EJECUT	LISREG	(lista de todos los registros)
	INCREG	(agrega un registro nuevo)
	MODREG	(modifica un registro existente)
	BORREG	(borra un registro)
	SAPROG	(guarda un archivo y da salida al programa)

Suponiendo que todos los módulos del programa, o la mayoría de ellos, ya se han elaborado, codificado en BASIC y verificado de forma individual, ¿cómo se pueden unir entre sí para conformar un programa completo? La mejor manera de afrontar este problema consiste en guardar cada módulo en cinta o en disco, otorgándoles el mismo nombre de archivo que empleáramos en las notas sobre el desarrollo del programa. De modo que INCREG se puede escribir y verificar en la mayor medida posible, guardándola después bajo el nombre de archivo INCREG. Normalmente, cuando se carga un programa desde cinta o disco, utilizamos la orden LOAD, seguida del nombre del archivo, como en LOAD "INCREG". Sin embargo, esto tiene el efecto de limpiar toda memoria, de modo que si cargáramos INCREG y posteriormente cargáramos MODREG, todo el programa INCREG desaparecería.

Por fortuna, existe una solución parcial. La orden MERGE permite cargar un programa desde cinta o disco sin borrar ningún otro programa que estuviera ya en la memoria. Pero para ello existe una condición importante. Si cualquiera de los números de línea del programa cargado mediante MERGE fueran iguales a los números de línea del programa que ya estuviera en la memoria, los nuevos se escribirían sobre los antiguos y se produciría un caos. Las versiones de BASIC que disponen de la orden RENUM pueden obviar este inconveniente volviendo a numerar las líneas de un módulo de programa antes de guardarlo, de modo que cuando se utilice MERGE no se plantee ningún problema.

Lamentablemente, en los ordenadores personales muchas versiones de BASIC no poseen la orden RENUM y, por lo tanto, será necesario efectuar una cuidadosa planificación de los números de línea desde el principio. Cuando se haya elaborado un diagrama de todos los módulos principales del programa (como hemos hecho parcialmente en la tabla), se podrán asignar los números de línea con los que se inicia cada módulo. En aquellas partes del programa en que con toda seguridad se introducirán cambios o modificaciones, como son el programa principal o las partes del programa donde se manipula el archivo, se deben efectuar incrementos de 50, o incluso de 100, en la numeración de las líneas, con el fin de dejar abundante espacio para los agregados. En los módulos del programa menos susceptibles de sufrir modificaciones, como la rutina PRESEN, pueden realizarse incrementos de 10 en la numeración de las líneas. Introducir un buen número de líneas REM en blanco en el programa no sólo contribuye a que éste resulte más fácil de leer, sino que también permite que posteriormente se

puedan agregar sentencias adicionales o llamadas a subrutinas. Si el BASIC de su ordenador tampoco dispusiera de MERGE, entonces tendrá que digitar los diversos módulos tal como están escritos y guardarlos juntos.

Los bloques del programa de la tabla se han fusionado (MERGE) como una "ejecución de prueba" en el listado impreso aquí, con el fin de ilustrar los peligros latentes del enfoque "pruebe y vea" que fomentan lenguajes como el BASIC. No tiene ningún sentido digitar el programa entero sólo para descubrir que no funcionará, pero si usted ha guardado todas las rutinas de anteriores capítulos, y si el BASIC de su ordenador posee la orden RENUM, puede volver a numerarlas (RENUM) y después fusionarlas (MERGE) para producir un listado similar.

El primer bloque del programa principal es INICIL, del cual se espera que inicialice variables, dimensione matrices, lea archivos, les asigne los datos a las matrices, establezca banderas, etc. La subrutina *INICIL* se divide en *CREMAT* (para crear matrices, *LEARCH* (para leer archivos y asignarles los datos de las matrices correspondientes) y *ESBAND* (para establecer banderas, etc.).

Cuando se ha efectuado todo esto, el programa pasa a *PRESEN*, una subrutina que imprime en la pantalla un mensaje de presentación. La última parte de esta rutina espera a que el usuario pulse la barra espaciadora para que el programa continúe.

El programa prosigue luego con *ELECCN*. Ésta consta de dos partes: la primera presenta un menú de las opciones que ofrece el programa de la agenda de direcciones; la segunda acepta la entrada de la opción desde el teclado y le asigna el valor (numérico) a una variable denominada OPCN.

El valor de esta variable lo utiliza el siguiente bloque del programa, EJECUT, para seleccionar uno de entre nueve bloques del programa. Todos ellos, a excepción de SAPROG, necesitarán retornar a *ELECCN* después de haberse ejecutado, para que el usuario tenga la oportunidad de seleccionar otra opción. Ello no será necesario en caso de seleccionarse 9 (SAPROG), porque se supone que esta opción pondrá fin a la operación del programa.

El principal problema de este programa, tal como está, es que el flujo de control no es correcto. INICIL insiste en que leamos un archivo con todos los datos, exista o no este archivo. Si el programa se está ejecutando por primera vez, no se habrá dado entrada a ningún registro y en la cinta o el disco no habrá archivos de datos. Todo intento de abrir y de leer un archivo inexistente tendrá como resultado un mensaje de error y el programa no funcionará.

Lo que se necesita es que la rutina *LEARCH* sea llamada sólo por uno de los módulos EJECUT, y entonces sólo una vez en cada ocasión se ejecutará el programa. Esto sugiere que debería haber una bandera ARCH, establecida originalmente en 0, que se establezca en 1 una vez que se haya leído el archivo. Si se hubiera establecido en 1, impediría todo intento ulterior de leerlo. INCREG entonces buscaría siempre a través de todas las matrices para localizar el primer elemento vacío y escribiría allí la información. Casi con toda certeza este registro no estaría entonces en la secuencia de clasificación correcta, de modo que debería haber una bandera RMOD que durante la ejecución se estableciera en 1. La bandera RMOD también se habría de establecer en 1 si se ejecutaran MODREG o BORREG. Puede intentar escribir el código correspondiente para conseguir esto; o, si simplemente desea ejecutar el programa, cambie la línea 1310 para que retorne (RETURN).

Tanto agregar como borrar o modificar un registro significa que es probable que la secuencia de registros esté desordenada, de modo que cualquier módulo (ENCREG, por ejemplo) debería primero verificar RMOD para ver si se han realizado cambios. De ser así, podríamos insistir en una clasificación antes de efectuar alguna búsqueda, o bien emprender una búsqueda estéril en una pila. SAPROG verificará RMOD y llamará a la rutina de clasificación si estuviera establecida (en 1) antes de guardar los datos del archivo en cinta o en disco.

Los aspectos de "interface humana" del programa, que hemos mencionado anteriormente, se pueden dividir en las siguientes categorías:

Interface para el usuario
Imagen para el usuario
Recuperación de errores
Seguridad
Adaptabilidad

Al hablar de *interface para el usuario* nos referimos a la forma en que el usuario del programa se comunica con éste. Hemos optado por utilizar menús (en lugar de órdenes). Muchas personas prefieren las órdenes, pero lo que importa es que, cualquiera sea la forma de intercomunicación empleada, ésta sea coherente. En el curso de un programa no debe haber órdenes similares que realicen cosas diferentes en distintas partes del mismo. De ser así, el usuario habría de leer atentamente cada menú antes de efectuar alguna selección y entonces no se podrían desarrollar "reflejos".

Tal como está conformado, nuestro programa es pobre en este sentido: el mensaje de presentación se termina pulsando la barra espaciadora; el menú de opciones se termina automáticamente digitando cualquiera de las teclas de 1 a 9; y, en el caso de INCREG, la entrada de datos se termina (para cada campo) pulsando la tecla RETURN. Esta clase de incoherencia podría ser aceptable en un programa "casero", pero no lo sería en el software comercial.

La *imagen para el usuario* se refiere a la forma en que el usuario percibe el funcionamiento del programa. En ella influye de manera decisiva la calidad de la interface para el usuario. La mayoría de las operaciones que se llevan a cabo en el interior del ordenador están totalmente ocultas al operador de la máquina. El único medio de que dispone para reconocerlas es la pantalla. Para el programa que estamos desarrollando, la imagen para el usuario más adecuada sería la representación de una agenda de direcciones verdadera. Del mismo modo, la imagen para el usuario más apropiada en el caso de un programa de tratamiento de textos sería la de un trozo de "papel" (en la pantalla) en el que se pudiera mecanografiar. En este caso, lo ideal sería que las palabras mecanografiadas en negrita aparecieran en negrita en la pantalla, que el texto subrayado se viera realzado y que los párrafos justificados se observaran así al visualizarlos.

La perfecta imagen para el usuario sólo se consigue en muy raras ocasiones (ninguna agenda de direcciones verdadera esperaría que el usuario pulsara 1 ("PRESS 1") para hallar un nombre. No obstante, una buena imagen para el usuario implica unos trazados de pantalla bien diseñados y un pa-

M N O P Q R S T U V W X Y Z

trón coherente para las operaciones. Los avisos siempre deberían aparecer en la misma posición en la pantalla (algunos procesadores de textos muy conocidos, por ejemplo, visualizan algunos avisos en la línea superior de la pantalla y otros en la línea inferior, aparentemente al azar). Un programa con una buena imagen para el usuario también debería informarle a éste, en cualquier momento que lo deseara, en qué punto del programa se halla. Si usted estuviera en la modalidad INCREG, por ejemplo, debería haber un mensaje siempre visible que se lo indicara. Si acabara de dar entrada a un campo (para agregarlo a un nuevo registro), debería haber un mensaje que le indicara, por ejemplo, PULSE RETURN SI LA ENTRADA ES CORRECTA, DE LO CONTRARIO PULSE ESCAPE (lo que nos lleva al importante tema de la recuperación e información de errores, que trataremos luego).

Lo ideal sería que todo el formato apareciera en la pantalla, de modo que, por ejemplo, el registro visualizado fuera del mismo formato que un registro impreso por la impresora. Muchos programas incorporan "menús de ayuda" que le indican cómo debe proseguir cuando no está seguro de ello.

Lo deseable es que la imagen para el usuario de un programa sea lo más concreta posible (un papel para mecanografiar o una tarifa, por ejemplo), evitando recurrir a objetos como "subarchivos", "buffers", etc. En este sentido, muchos programas comerciales de base de datos no son todo lo eficientes que cabría esperar; el usuario ha de tener presente en todo momento que ciertas informaciones se encuentran en archivos o en campos ocultos y temporales. Estos factores tienden a transformar la utilización de un programa de estas características en un arduo esfuerzo intelectual.

La *recuperación de errores* es también un tema importante. ¿Qué sucedería, por ejemplo, si acabara de dar entrada al nombre de alguien y de inmediato se percatara de que ha cometido un error de digitación? ¿Tendría que seguir adelante y después llamar a MODREG para corregirlo, o el programa le ofrecerá la posibilidad de "abandonar" antes de ir más lejos? La mayoría de las versiones de BASIC informarán de los errores al dar entrada a un programa, ya sea cuando se da entrada a la línea errónea o cuando se ejecuta el programa. Sin embargo, esto no forma parte de la "interface para el usuario". El BASIC incluye varios mensajes que vuelven a solicitarle al usuario una entrada correcta en caso de que se hubiera realizado una errónea (p. ej., el aviso REDO —rehacer— cuando se hace una entrada inadecuada en respuesta a una sentencia INPUT).

La manipulación de errores posee dos facetas: la información sobre el error y la recuperación del error. Un programa muy conocido de tratamiento de textos, por ejemplo, posee una buena información de errores pero una pobre recuperación de ellos; si después de crear un largo documento usted intentara guardarlo en un disco que estuviera casi lleno, el programa le proporcionaría un útil mensaje: AGOTADO ESPACIO DISCO. Lamentablemente, no le permite al usuario recuperarse de este error: ¡no se puede dar formato a un disco nuevo sin destruir primero el texto que quizá le haya costado horas digitar!

Toda operación que pueda efectuar el usuario y que pudiera producir pérdida de datos (MODREG, por ejemplo) debería siempre ser objeto de alguna indagación antes de ser ejecutada. Siempre se deberían proporcionar mensajes como ESTO DESTRUIRA EL REGISTRO, ¿ESTA USTED SEGURO? (S/N). En el caso de un procesador de textos, un mensaje de esta índole sería: LA ORDEN "SAVE" NO CONSERVARA UNA COPIA DEL DOCUMENTO ANTIGUO. ¿LE PARECE BIEN? (S/N).

La manipulación de errores (detectarlos e informar de que se han producido) se debe tener en cuenta al diseñar un programa siempre que exista la posibilidad de una entrada de datos incorrecta, de una opción del menú equivocada, de órdenes erróneas y siempre que la información se haya de modificar o de guardar, especialmente si guardarla implica tener que escribir sobre datos antiguos.

El usuario también debe prestar atención a la *seguridad*: ¿qué sucedería con el programa o con los datos si se produjera un grave contratiempo (p. ej., un corte del fluido eléctrico)? Un procesador de textos bastante sofisticado incorpora un programa denominado RECOVER (recuperar), que, en el caso de que se produzca un grave percance (p. ej., un corte del fluido eléctrico, o que el usuario apague el ordenador sin antes haber guardado el documento), no permite que se pierda casi nada. No obstante, este tipo de técnicas avanzadas de programación se encuentran más allá de las metas de nuestro curso. Lo importante es lograr que sus programas sean lo más seguros posible, anticipándose a los errores garrafales en que se pueda incurrir —neutralizándolos de manera racional— y escribiendo rutinas concebidas para hacerles frente.

La *adaptabilidad*, es decir, la facilidad con que el usuario se puede familiarizar con el programa, también es importante. Por norma general, siempre se debe dejar abundante espacio entre los números de línea (en BASIC) e incorporar muchas líneas REM en blanco, donde posteriormente, en caso de ser necesario, se puedan incluir sentencias y GOSUB. Al crear matrices, se debe incorporar al menos una matriz innecesaria para posibilitar una futura ampliación. Una regla fundamental de la escritura de programas es que las exigencias futuras no se pueden anticipar. Lo único seguro es que un buen programa siempre se puede mejorar.

Complementos al BASIC

SPECTRUM

Véase el programa de la página 318 para la versión del Spectrum desde la línea 1300 a la 1370. El bucle del programa principal entre las líneas 3750 y 3780 funcionará en el Spectrum, pero podría crear problemas debido a que las teclas de este ordenador se repiten si se mantienen pulsadas durante más de una fracción de segundo. El manual del Spectrum recomienda que, para evitar este inconveniente, este tipo de bucle INKEY$ incluya una línea extra:

```
3755 IF INKEY$ <> "" THEN GOTO 3755
```

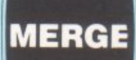
El Spectrum admite la función VAL(A$), pero romperá el programa si el primer carácter de A$ no es numérico; en este programa el problema se puede obviar mediante:

```
3790 LET OPCN = CODE A$ - 48
```

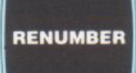
pero esta solución no es perfecta, funciona sólo cuando A$ es un carácter único (como debe ser en el programa). El Spectrum no admite ON...GOSUB, pero sí le permite escribir GOSUB (expresión numérica) así como sencillamente GOSUB (número de línea); la línea 4010 se puede reemplazar por:

4010 GOSUB (290+OPCN*20)

El Spectrum no admite RENUMBER



No la admiten el Oric-1, el Commodore 64, el Vic-20, el Lynx, el Dragon 32 y el BBC Micro. No obstante, a menudo suele haber una forma de imitar la orden MERGE, que le podrían revelar algún usuario o alguna publicación especializada. El Lynx posee la orden APPEND, que le permite LOAD (cargar) un programa en el final de un programa de la memoria; esta orden puede reemplazar a MERGE siempre que el primer número de línea del programa en cinta sea mayor que el último número de línea del programa de la memoria.



No la admiten el Spectrum, los Commodore y el Oric-1; en estas máquinas la renumeración se puede realizar línea a línea utilizando la instrucción EDIT.

INKEY$

Véase "Complementos al BASIC" de página 175.

OPEN CLOSE

Véase "Complementos al BASIC" de página 319.



Reemplazarla por CLS, excepto en el Commodore 64 y el Vic-20; en estas máquinas digite PRINT "shiftkey+CLRkey", que hará que entre las comillas aparezca un corazón de campo invertido o una S mayúscula.

ON... GOSUB

En la línea 4010, ON...GOSUB hará que el control pase a líneas inexistentes (310, 330, 350, etc.), lo que provocará la ruptura del programa. Estas líneas posteriormente se ampliarán para incluir las subrutinas para ejecución del menú; mientras tanto, dé entrada en el programa a lo siguiente, a modo de líneas "ficticias":

```
310 RETURN
330 RETURN
```

etc.

STR$

El Lynx no admite esta orden, de modo que reemplace la línea 9080 por:

```
9075 N = TAMA
9077 GOSUB 9500
9080 INDCAM$ (TAMA) = N$
```

e inserte esta subrutina:

```
9500 REM Para convertir N en N$, donde N
     es un entero
9510 LET N$ = " "
9520 IF N<0 THEN LET N$=""
9530 N=ABS(N)
9540 X=10
9550 I=1
9560 FOR K=1 TO 8
9570 I=I*X
9580 R=K
9590 If I>N THEN K=8
9600 NEXT K
9610 FOR K=1 TO R
9620 I=I/X
9630 Z=INT(N/I)
9640 LET N=N-Z*I
9650 N$=N$+CHR$(48+INT(Z))
9660 NEXT K
9670 RETURN
```

```
10 REM "PROPRI"
20 REM *INICIL*
30 GOSUB 1000
40 REM *PRESEN*
50 GOSUB 3000
60 REM *ELECCN*
70 GOSUB 3500
80 REM *EJECUT*
90 GOSUB 4000
100 END
1000 REM SUBRRUTINA *INIC*
1010 GOSUB 1100: REM  SUBRRUTINA *CREMAT* (CREAR MATRICES)
1020 GOSUB 1300: REM  SUBRUTINA *LEARCH* (LEER ARCHIVOS)
1030 GOSUB 1380: REM  SUBRUTINA *ESBAND* (ESTARLECER BANDERAS)
1040 REM
1050 REM
1060 REM
1070 REM
1080 REM
1090 RETURN
1100 REM SUBRUTINA *CREMAT* (CREAR MATRICES)
1110 DIM NOMCAM$(50)
1120 DIM MODCAM$(50)
1125 DIM CLLCAM$(50)
1130 DIM CIUCAM$(50)
1140 DIM PROCAM$(50)
1150 DIM TELCAM$(50)
1160 DIM INDCAM$(50)
1170 REM
1180 REM
1190 REM
1200 REM
1210 LET TAMA = 0
1220 LET RMOD = 0
1230 LET SVED = 0
1240 LET CURS = 0
1250 REM
1260 REM
1270 REM
1280 REM
1290 RETURN
1300 REM SUBRUTINA *LEARCH* -- VEASE RECUADRO "COMPLEMENTOS"
1310 ON ERROR GOTO 1370
1320 OPEN "I",#1,"DAT.AGCO"
1330 FOR L = 1 TO 50
1340 INPUT #1 NOMCAM$(L),CLLCAM$(L),CIUCAM$(L),PROCAM$(L), TELCAM$(L)
1350 NEXT L
1360 CLOSE #1
1370 RETURN
1380 REM RUTINA *ESBAND* FICTICIA
1390 RETURN
3000 REM SUBRUTINA *PRESN*
3010 PRINT
3020 PRINT
3030 PRINT
3040 PRINT
3050 PRINT TAB(11);"*BIEN VENIDO A LA *"
3060 PRINT TAB(9);"*AGENDA COMPUTERIZADA*"
3070 PRINT TAB(12);"*DE MI COMPUTER*"
3080 PRINT
3090 PRINT TAB(0);"(PULSE BARRA ESPACIADORA PARA CONTINUAR)"
3100 FOR L = 1 TO 1
3110 IF INKEY$ <> " " THEN L = 0
3120 NEXT L
3130 PRINT CHR$(12)
3140 RETURN
3500 REM SUBRUTINA *ELECCN*
3510 REM
3520 REM "IMMENU"
3530 PRINT CHR$(12)
3540 PRINT "SELECCIONE UNA DE LAS SIGUIENTES OPCIONES"
3550 PRINT
3560 PRINT
3570 PRINT
3580 PRINT "1. HALLAR REGISTRO (DE NOMBRE)"
3590 PRINT "2. HALLAR NOMBRES (DE NOMBRE INCOMPLETO)"
3600 PRINT "3. HALLAR REGISTRO (DE CIUDAD)"
3610 PRINT "4. HALLAR REGISTRO (DE INICIAL)"
3620 PRINT "5. LISTAR TODOS LOS REGISTROS"
3630 PRINT "6. AGREGAR REGISTRO NUEVO"
3640 PRINT "7. MODIFICAR REGISTRO"
3650 PRINT "8. BORRAR REGISTRO"
3660 PRINT "9. SALIR Y GUARDAR"
3670 PRINT
3680 PRINT
3690 REM  "ASOPCN"
3700 REM
3710 LET L = 0
3720 LET I = 0
3730 FOR L = 1 TO 1
3740 PRINT "DE ENTRADA A OPCION (1 - 9)"
3750 FOR I = 1 TO 1
3760 LET A$ = INKEY$
3770 IF A$ = "" THEN I = 0
3780 NEXT I
3790 LET OPCN = VAL(A$)
3800 IF OPCN < 1 THEN L = 0
3810 IF OPCN > 9 THEN L = 0
3820 NEXT L
3830 RETURN
4000 REM SUBRUTINA *EJECUT* -- VEASE RECUADRO "COMPLEMENTOS"
4010 ON OPCN GOSUB 310, 330, 350, 370, 390, 410, 430, 450, 470
4020 RETURN
9000 REM SUBRUTINA *INCREG*
9010 PRINT CHR$ (12)
9020 INPUT "DE ENTRADA AL NOMBRE";NOMCAM$(TAMA)
9030 INPUT "DE ENTRADA A LA CALLE";CLLCAM$(TAMA)
9040 INPUT "DE ENTRADA A LA CIUDAD" CIUCAM$(TAMA)
9050 INPUT "DE ENTRADA A LA PROVINCIA" PROCAM$(TAMA)
9060 INPUT "DE ENTRADA AL NUMERO DE TELEFONO";TELCAM$(TAMA)
9070 LET RMOD = 1
9080 LET INDCAM$(TAMA) = STR$(TAMA)
9090 GOSUB *MODNOM*
9100 RETURN
```

Tony Lodge

Un buen sonido

La generación de sonido en el BBC Modelo B

Unas excepcionales configuraciones para sonido y amplias órdenes en BASIC para controlarlas hacen que el BBC Micro sea uno de los mejores ordenadores para aquellos usuarios que se interesen, sobre todo, por el sonido. Se proporcionan como estándar tres osciladores de onda cuadrada independientes, ocho tipos de ruido y cuatro envolventes independientes de ADSR (arranque, demora, sostenido, liberación) y tono. Ello significa que se pueden construir frases musicales de hasta tres voces en armonía; además, órdenes especiales permiten que las notas seleccionadas para formar un acorde suenen exactamente al mismo tiempo.

Creación del sonido

En su forma más simple, el sonido se puede crear mediante la orden SOUND:

```
SOUND C,V,P,D
```

En este caso:

C = Número del canal u oscilador (0-3)
V = Volumen
P = Tono de la nota
D = Duración o longitud de la nota

Cualquiera de los tres osciladores (1, 2 y 3) puede tocar una nota; 0 corresponde a ruido. El volumen adopta un valor entre 0 (apagado) y −15 (fuerte). El tono se define en intervalos de un cuarto de nota (medio semitono) entre 0 (*la #* a 116,5 Hz) y 255 (*re* a 4698,64 Hz), ofreciendo una escala de cinco octavas y media de extensión. El do_3 (o central) se establece mediante el valor 28. Si se ha especificado el canal de ruido, existen ocho tipos disponibles, que determinan los siguientes números de tono:

Número	Tipo de ruido
0	Trémolo de tono agudo
1	Trémolo de tono medio
2	Trémolo de tono grave
3	Trémolo. El tono varía con el tono del canal 1
4	Tono agudo
5	Tono medio
6	Tono grave
7	El tono varía con el tono del canal 1

Por último, la duración de la nota se controla en intervalos de veinteavos de segundo entre 1 y 255, es decir, una nota puede durar hasta 12,75 segundos. Una orden para tocar el la_3 en el canal 1 a un volumen medio de −7 durante medio segundo se construye de la siguiente forma:

```
SOUND 1,−7,46,10
```

Si el ordenador recibiera una segunda orden SOUND antes de haber completado la primera, la nota se colocaría al final del canal.

Además de las funciones sencillas que hemos re-

Construcción de luz

Las facilidades para gráficos del Commodore 64

COLOR DEL FONDO

MULTI-COLOR 1

MULTI-COLOR 2

COLOR DEL PRIMER PLANO

El ordenador Commodore 64 proporciona gran cantidad de posibilidades al programador de gráficos. No obstante, adolece de un inconveniente, que comparte con el Vic-20: su muy limitado juego de órdenes estándar en BASIC. Mediante las instrucciones POKE y PEEK el programador consigue acceder a todas las configuraciones de la máquina, pero algunos de los procedimientos necesarios pueden resultar bastante confusos. Asimismo, al igual que en el Vic-20, existen varios caminos para abrirse paso a través de este laberinto. Y éstos adoptan la forma de varios paquetes de programas producidos por fabricantes independientes que simplifican muchísimo la creación de sprites o de caracteres definidos por el usuario. Además de este software, Commodore fabrica su propio cartucho enchufable, denominado *Simon's* BASIC.

El Commodore 64 ofrece sus juegos de caracteres estándar en mayúscula y minúscula en modalidades de visualización normal o invertida. También se encuentran a la venta los caracteres especiales para gráficos que originariamente se desarrollaron para el PET y que también se utilizan con el Vic-20. La visualización en pantalla consta de 40 columnas y 25 filas y los caracteres aparecen en ella imprimiéndolos (PRINT) en cualquier color elegido o colocando (POKE) los códigos correspondientes en la memoria de pantalla y de color detalladas en el manual del usuario. El juego de caracteres especiales para gráficos del Commodore constituye un instrumento muy adaptable, con el cual se pueden unir atractivas visualizaciones en baja resolución. En esta tarea es posible emplear más de 60 caracteres especiales, que se puede visualizar en modalidad normal o invertida, dando al usuario más de 120 posibilidades de elección al diseñar una figura.

Si no se consigue hallar el carácter exacto, entonces se pueden definir caracteres nuevos para utilizar en la visualización. Esto, lamentablemente, no es muy sencillo de hacer. Para definir un carácter el programador debe, primero, copiar, de la memoria ROM a la RAM, todos los caracteres normales que se hayan de emplear, antes de colocar (POKE) los números necesarios en la serie de posiciones que habrán de retener el carácter nuevo.

señado, la capacidad de la orden SOUND se puede ampliar modificando su formato:

SOUND & HSFC,V,P,D

En este caso el signo & instruye al ordenador para que trate HSFC como un número hexadecimal de cuatro dígitos (véase p. 179). Sin embargo, para comprender cómo se debe utilizar la orden modificada únicamente es necesario analizar la función de cada dígito. Sólo tres de éstos son significativos, ya que C es simplemente el número de canal normal descrito en SOUND.

H puede estar sin especificar (0) o bien establecida en 1. H = 1 le permite a la nota anterior del mismo canal continuar hasta el final. Por otra parte, si se construyera una nota con una fase de liberación larga (apagándose lentamente), el ordenador supondría erróneamente que la nota ha finalizado mientras todavía está sonando y comenzaría de manera abrupta la nota siguiente en mitad de ella. H establecida en 1 obliga al ordenador a esperar que se complete. Cuando se utiliza el parámetro H, las restantes partes de la orden SOUND & se ignoran, a excepción del número de canal.

S le permite al usuario especificar un número de notas a ejecutar al mismo tiempo en distintos canales, creando un acorde. S = 0 deja la orden en su estado normal. S se establece en 1 si se requiere que al mismo tiempo se toque una nota en alguno de los otros canales. Si se establece en 2, se tocarán notas en los otros dos canales. En efecto, cuando el ordenador se encuentra en un valor S retiene la nota correspondiente hasta que puede dar cuenta de la otra nota o las otras dos notas relacionadas del acorde que están indicadas con el mismo número S en los canales restantes. Entonces ejecuta juntas todas las notas especificadas.

F se establece en 0 o en 1. Cero no tiene ningún efecto, pero 1 hace que el ordenador descarte todas las notas que esperan ser ejecutadas al final del canal, interrumpe la ejecución de la nota en curso y toca inmediatamente la nota contenida en su propia orden.

La mejor forma de ilustrar SOUND & es mediante un ejemplo. Este programa toca la primera línea de *Happy birthday to you* (Cumpleaños feliz) en la tonalidad de *fa#* (notas inmediatamente superiores al do_3):

fa# fa# sol# fa# si la#:

También incluye un acorde mayor de tres notas para el *la#* final compuesto por *re, fa* y *la#*

```
10 SOUND 1,–7,40,20: REM *1ER FA#*
20 SOUND 1,–7,40,10: REM *2DO FA# BREVE*
30 FOR I = 1 TO 3: READ N
40 SOUND 1,–7,N,20: NEXT I: REM *SOL#
   FA# SI*
50 SOUND &201,–7,32,30: REM *RE*
60 SOUND &202,–7,38,30: REM *FA*
70 SOUND &203,–7,48,30: REM *LA#*
80 DATA 44,40,50
90 END
```

Las capacidades de sonido que proporciona la orden ENVELOPE las analizaremos en un próximo capítulo.

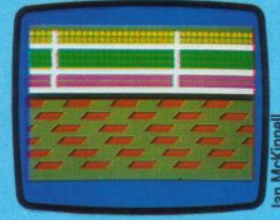
Cada celda de caracteres se compone de líneas de puntos pixel de ocho por ocho, que se representan en forma binaria como grupos de unos y ceros. Normalmente 1 se interpreta como un pixel establecido para el color del primer plano o del carácter, y 0 se interpreta como un pixel establecido para el color del fondo o de la pantalla. El Commodore 64 proporciona la posibilidad de representar hasta cuatro colores dentro de cada celda de caracteres. Esto se conoce como modalidad multicolor. Cuando el ordenador se coloca en esta modalidad, utiliza dos bits para representar el color de cada pixel.

Para cualquier par de bits existen cuatro combinaciones posibles, que se emplean para representar cuatro colores diferentes en el interior de la celda de caracteres. Cada una de las celdas de caracteres de la pantalla se puede establecer para que se la interprete de forma normal o bien como un carácter multicolor, pero en este último caso la elección de colores disponibles se reduce de 16 a 8. Existen otros inconvenientes: los dos bits multicolores han de ser los mismos para todos los caracteres de la pantalla; además, en la modalidad multicolor la resolución horizontal se reduce a la mitad.

En el BASIC de Commodore no existen órdenes para gráficos de alta resolución. No obstante, se pueden crear visualizaciones de igual resolución utilizando la técnica conocida como *confección de mapas de bits*. La visualización en pantalla del Commodore 64 consta de 64 000 pixels. La confección de mapas de bits se efectúa posibilitando que el usuario encienda o apague cada uno de los pixels de forma individual. Se trata de un procedimiento bastante complicado como para que lo pueda realizar el usuario corriente; por otra parte, al utilizar el BASIC de esta manera, la imagen en alta resolución se va creando con notable lentitud. En la práctica, existen dos alternativas para quienes deseen crear gráficos de alta resolución: la primera es comprar el cartucho *Simon's* BASIC de Commodore; la segunda, aprender a programar en código de lenguaje máquina. La modalidad estándar de alta resolución del Commodore 64 divide la pantalla en 200 líneas de 320 pixels. También se puede adquirir modalidad multicolor de alta resolución.

A continuación transcribimos un breve programa que utiliza el juego de caracteres Commodore para crear una imagen de un supermercado. En un próximo capítulo analizaremos los sprites y el *Simon's* BASIC.

Primer paso
Esta escena estática de supermercado se creó utilizando gráficos de baja resolución. En un próximo capítulo de esta sección agregaremos, utilizando gráficos sprite, un comprador con un carrito móvil


Ian McKinnell

```
3000 REM ** SUPERMARKET **
3010 PRINT"■"
3020 POKE53280,6:POKE53281,12
3030 PRINT"■●●●●■ ■●●●●●●●●●●●●●●●●●●●●●●●■ ■●●●●●●●●●●●●●●"
3040 PRINT"■OOOO■ ■OOOOOOOOOOOOOOOOOOOOOOO■ ■OOOOOOOOOOOOOO"
3050 PRINT"■                                      ■"
3060 PRINT"■◆◆◆◆■ ■◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆■ ■◆◆◆◆◆◆◆◆◆◆◆◆◆◆"
3070 PRINT"■◆◆◆◆■ ■◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆■ ■◆◆◆◆◆◆◆◆◆◆◆◆◆◆"
3080 PRINT"■◆◆◆◆■ ■◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆■ ■◆◆◆◆◆◆◆◆◆◆◆◆◆◆"
3090 PRINT"■                                      ■"
3100 PRINT"■♥♥♥♥■ ■♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥■ ■♥♥♥♥♥♥♥♥♥♥♥♥♥♥"
3110 PRINT"■♥♥♥♥■ ■♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥■ ■♥♥♥♥♥♥♥♥♥♥♥♥♥♥"
3120 PRINT"■                                      ■"
3130 PRINT"■                                      ■"
3140 PRINT"■  ▼   ▼   ▼   ▼   ▼   ▼   ▼   ▼   ▼"
3150 PRINT
3160 PRINT" ■▼   ▼   ▼   ▼   ▼   ▼   ▼   ▼   ▼   ▼"
3170 PRINT
3180 PRINT"▼   ▼   ▼   ▼   ▼   ▼   ▼   ▼   ▼   ▼  "
3190 PRINT
3200 PRINT"■ ▼   ▼   ▼   ▼   ▼   ▼   ▼   ▼   ▼   ▼
3210 PRINT
3220 PRINT"■  ▼   ▼   ▼   ▼   ▼   ▼   ▼   ▼   ▼"
3230 PRINT
3240 PRINT" ▼   ▼   ▼   ▼   ▼   ▼   ▼   ▼   ▼   ▼"
3250 FORI=1063TO2023STEP40
3260 POKEI,160:POKE54272+I,6:NEXT
3270 GOTO3270
```

Torres Quevedo

Además de construir dirigibles y transbordadores, este científico español contribuyó decisivamente al desarrollo de la informática

Leonardo Torres Quevedo, el científico que aplicó por primera vez la aritmética de coma flotante a los ordenadores, nació en Santa Cruz (Cantabria) el 28 de diciembre de 1852. Estudió en el Instituto de Bilbao y en la Facultad de Ingeniería de Madrid.

Su genio inventor alcanzó su punto culminante en los últimos años de su vida. A él se le debe la construcción del transbordador funicular aéreo junto a las cataratas del Niágara, que en la actualidad continúa en uso; además diseñó un dirigible dotado de una armadura funicular que reunía las propiedades de los dirigibles rígidos y flexibles. Pero, básicamente, Torres era un hijo de su época y su principal interés se centraba en los dispositivos electromecánicos. En 1906, en Bilbao, exhibió ante el rey de España, Alfonso XIII, el prototipo de un barco dirigido a distancia por medio de las ondas hertzianas, y en 1911 inventó el primer jugador de ajedrez autómata. Para mover las piezas la máquina utilizaba unos electroimanes situados debajo de la mesa; estaba programada para ganarle una partida sencilla a un oponente humano.

El interés de Torres por los autómatas nació a raíz de su experiencia en las líneas de montaje de las plantas industriales de la Europa de comienzos del siglo xx; y a lo largo de toda su vida se esforzó por separar los tipos de problema que requerían pensamiento humano para su resolución de aquellos que se podían resolver automáticamente.

Científico polifacético
Al igual que muchos de sus contemporáneos, Torres Quevedo fue un científico que dedicó sus esfuerzos a la investigación en diferentes campos. Sus intereses abarcaron desde el diseño y construcción de dispositivos mecánicos como el dirigible que vemos en la fotografía hasta la creación de máquinas de calcular electromecánicas, ya dentro de los dominios de la matemática pura



Cortesía de la Royal Aeronautical Society

En 1914 publicó un estudio en el que demostraba la viabilidad de la construcción del ingenio analítico de Babbage (véase p. 220) empleando técnicas electromecánicas, y fue en este documento donde sugirió por primera vez la utilización de la aritmética de coma flotante para todo futuro ordenador. En 1920 construyó una calculadora electromecánica que utilizaba una máquina de escribir adaptada para dar entrada a los números e imprimía los resultados automáticamente. La máquina de escribir estaba conectada a la calculadora mediante cables telefónicos, y Torres previó la posibilidad de acoplar muchos terminales a una calculadora central (o unidad de proceso).

Torres Quevedo fue condecorado por la Academia Francesa de Ciencias, y con posterioridad sería nombrado presidente de honor vitalicio de la Academia de Ciencias de Madrid. Falleció en Madrid el 18 de diciembre de 1936.

La aritmética de coma flotante

Una caja registradora visualiza los totales en pesetas y céntimos (12,50, por ejemplo) y en una máquina de este tipo sólo son necesarios dos espacios después de la coma decimal. Pero en un ordenador, con frecuencia se requiere mayor exactitud y se permite que el número de lugares decimales varíe o "flote" de acuerdo a las exigencias del problema. Esto se conoce como "aritmética de coma flotante".

Cualquier número se puede escribir de diversas maneras. Por ejemplo, 0,8752 metros se puede expresar como 875,2 milímetros, o $0,8752 \times 1000$ milímetros, o simplemente $0,8752 \times 10^3$. Este último procedimiento se presta para cumplir las funciones de codificación en un ordenador. Si éste sólo tiene asignados seis espacios digitales para representar a cada número (y, en aras de la claridad, se utiliza el sistema decimal en lugar del binario), entonces el número anterior se puede almacenar como 875203: donde los dos últimos dígitos de la derecha se denominan *índice* y representan la potencia de 10 (en este caso, 3) y los cuatro primeros dígitos reciben el nombre de *mantisa*. Para dar otro ejemplo para este ordenador: el número 418302 representa $0,4183 \times 10^2$, o 41,83.

La mantisa y el índice generalmente se "normalizan" para eliminar de la mantisa los primeros ceros. Por ejemplo, el número 41,83 se podría escribir como 004104, pero se normalizaría en 418302 (incluyendo, por lo tanto, más dígitos significativos en la mantisa).

La forma índice-mantisa de la aritmética de coma flotante ofrece la ventaja de que se puede representar una amplia escala de números. En el caso del ordenador que mencionábamos anteriormente, que dispone de sólo dos dígitos para el índice, puede tratar con un número tan elevado como $0,9999 \times 10^{99}$, o con un número tan pequeño que posea 98 ceros después de la coma decimal y antes del primer dígito distinto a cero.

No obstante, la exactitud de este sistema permanece limitada a los dígitos que se destinen a la mantisa. Por consiguiente, algunos números sólo se pueden representar aproximadamente y para evitar que se produzcan errores las técnicas de la programación aritmética se han de asumir con gran cuidado. Ésta es la razón por la cual, en algunos ordenadores, (1/3)*3 dará como resultado 0,9999999, en vez de la respuesta verdadera, que es 1.

UNION PERFECTA

Así se comportan los periféricos creados por SINCLAIR para SINCLAIR: de forma perfecta. Y es lógico.

Cada vez que SINCLAIR diseña un microordenador, no lo hace de una manera aislada. Simultáneamente crea todos esos periféricos que van a hacer más potente, preciso y útil el microordenador que tiene entre manos.

Periféricos pensados y diseñados para dar un servicio óptimo, pero con un precio razonable, dentro de la filosofía SINCLAIR:

"Hacer la informática accesible a todos".

Por eso cuando creó el ZX 81 vio la necesidad de dotarlo con una ampliación de memoria de 16K RAM para que no quedara pequeño y de una impresora sencilla y barata pero útil y precisa.

Así es la filosofía SINCLAIR. Así son los periféricos de SINCLAIR para SINCLAIR.

Microordenadores

sinclair

Toda una filosofía.



CENTRAL COMERCIAL: Tomás Bretón, 60
Tel. 468 03 00 Telex: 23399 IYCO E Madrid.
DELEGACION CATALUÑA: Camp, 80 - Barcelona - 22

PARA JUGAR A LO GRANDE
(INSTANTANEAMENTE)

Presentamos **el Interface 2 ZX** Pensado y diseñado por SINCLAIR para unirse a la perfección con tu microordenador Spectrum.

Si a la hora de elegir tu microordenador optaste por el mejor, es lógico que elijas ahora el Interface 2 ZX

Ya habrás podido deleitarte con la más amplia variedad de juegos existentes para tu Spectrum (la más extensa del mercado). Ahora con el Interface 2 ZX vas a tener más ventajas para tu Spectrum:

- Podrás conectar Joysticks para sacarle, aún, mayor rendimiento a tus mejores juegos y divertirte con aquellos exclusivamente disponibles en **Cartuchos ZX:** correr, saltar, volar... a lo grande.
  ¡Menuda diferencia!
- Además, al ser cartuchos con memoria ROM, podrás, con tu SPECTRUM de 16 K, jugar con programas hasta ahora reservados para 48 K, sin ampliar la memoria.
  ¡Vaya ahorro!
- Al conectar el Interface 2 ZX tienes la certeza de poseer un periférico pensado por SINCLAIR para SINCLAIR. Tu microordenador queda a salvo de circuitos poco fiables.
  ¡Un alivio!
- Al adquirir el Interface 2 ZX y los Cartuchos ZX en la red de Concesionarios Autorizados, podrás exigir la tarjeta de garantía INVESTRONICA, única válida en territorio nacional.
  ¡Una tranquilidad!



Interface 2 ZX y Cartuchos ZX

Si aún no los tienes
no sabes lo que te pierdes

Solicita una demostración en cualquier Concesionario Autorizado INVESTRONICA.

